
La primera revista para profesionales del diseño por ordenador
3D WORLD
ARGENTINA 10 $ • CHILE 3000 $ • PORTUGAL 1250 ESC (CONT)
AÑO 2 • NÚMERO 15 • P.V.P. 995 PTAS.
CD ROM PC/MAC:
640Mb
Demos de Animatek World Builder, Serif PagePlus, JetaReyes, Organica, Vertigo 3D Words, 3D Web Workshop, PhotoFix, CyberStudio. Edición de vídeo con Smacker. Brushmaps, Efectos, Objetos y Texturas para Imagine. IPAS para 3D Studio. Plug-Ins para 3D MAX, Lightwave y Real 3D. 37 objetos 3DS y 72 modelos VRML. Ejemplos de los artículos y creaciones de los lectores.
Smacker, la alternativa multiplataforma al formato AVI
Modelado Orgánico Dota a tus personajes de un aspecto más real
CURSOS: POV Directivas condicionales y funciones • 3D STUDIO Opciones de Render • TRUESPACE Las funciones de animación • IMAGINE Primeros pasos en el Action Editor • LIGHTWAVE Utilizando el entorno de trabajo • REAL 3D Operaciones con B-Splines • SOFTIMAGE Las curvas de función • STRATA STUDIO PRO Animación avanzada
WORKSHOP PROGRAMACIÓN: Optimizando OpenGL
NOMBRES PROPIOS: Rebeca Allen, el cuerpo sintético
RECREACIÓN DE ATMÓSFERAS: Lluvia con 3D MAX (II)
MAYA
El secreto mejor guardado ve la luz
PC • MAC • AMIGA • SGI
Prensa Técnica
00015
8 413042 087731



Edita PRENSA TÉCNICA S.L.

Director/Editor
Mario Luis

Coordinador Técnico
Miguel Cabezuelo

Edición
Charo Sánchez

Colaboradores
Rafael Morales, Antonio Casado, Enrique Urbaneja, Miguel Ángel Díaz Martín, Jesús Nuevo, Guillermo Gómez, Christian Daniel Semczuk, César M. Vicente, Antonio Marchal, José María Ruiz, David Díaz González, Miguel Ángel Díaz Aguilar, Bruno de la Calva, Fernando Cazaña, Julio Martín Erro, Juan Carlos Olmos

Asesor Técnico
Eduardo Toribio

Jefa de Maquetación
Carmen Cañas

Diseño y Maquetación
Manuel J. Montes
Marga Vaquero
Pedro Bustos

Portada
Carlos Sánchez

Imágenes de Portada
Cedidas por SGO

Publicidad
Marisa Fernández

Suscripciones
soniag@prensatecnica.com

Filmación
M Y F

Impresión
Printerman Industria Gráfica

Duplicación del CD-ROM
M.P.O.

Distribución
SGEL

Redacción, Publicidad y Administración
C/ Alfonso Gómez, 42
Nave 1-1-2
28037, MADRID, ESPAÑA
Telf.: (91) 304 06 22
Fax: (91) 304 17 97

Depósito legal: M-2075-1997
ISSN: 1137-3970

AÑO 2 • NÚMERO 15
Copyright 30/06/1998

PRINTED IN SPAIN

MAYA, UN NUEVO COMIENZO

EDITORIAL

Por fin se presentó, después de tanto tiempo. La verdad es que antes de que comenzara la presentación oficial de Maya, todo era expectación. Los allí presentes estaban deseando saber con qué sorprendería la nueva herramienta de Alias|Wavefront, tan anunciada y esperada durante meses.

Y a medida que transcurría la demostración, la gente se maravillaba, pero se mostraba un poco recelosa, y con motivo, pues no es lo mismo ver las evoluciones de un software en una demostración que hace alguien, que comprobarlo en el trabajo, en la máquina, buceando en el interfaz y "metiendo mano" allí donde deseemos. Es decir, en la práctica, la parte más importante de todo programa de modelado y animación.

Lo cierto es que, aunque no sea lo mismo ver que tocar, el programa promete, y mucho, algo esperado tratándose de Alias|Wavefront. Pero, a pesar de ello, no nos aventuramos a decir que estamos ante la revolución, o pensar que no merece la pena gastarse una millonada en una O2 con Maya, precisamente por eso, porque no hemos "jugado" con él. Sabemos que el producto ha tenido un proceso de desarrollo largo, se ha mantenido casi todo lo conocido de él en el máximo de los secretos, y eso tal vez quiera decir que nos quieren sorprender con algo grande. Por eso, dentro de poco prepararemos una evaluación de Maya y os comentaremos todos y cada uno de sus secretos.

Por ello, la presentación de Maya ocupa el puesto destacado de nuestra portada, contándoos cómo fue el evento porque estaréis deseando saber lo que allí aconteció. También, seguimos con los reportajes dedicados a la animación de personajes sintéticos y la recreación de atmósferas en 3D MAX, dos temas de máximo interés. Asimismo, os presentamos el nuevo formato de vídeo multiplataforma que está dispuesto a plantear una dura batalla a los estándares AVI, QuickTime o MPEG. Se trata del SMK, nombre derivado de la herramienta de compresión Smacker, que será analizada en nuestras páginas. Su sencillez de uso, su portabilidad a casi todas las plataformas, la posibilidad de generar ficheros ejecutables con el vídeo deseado y, sobre todo, el hecho de que sea *Freeware*, hacen de este formato una opción a tener muy en cuenta, y seguro que su popularidad no tardará en llegar. Además, nuestros correspondientes cursos, trucos, secciones de software y hardware, y un interesante artículo dedicado al modelado orgánico. Para finalizar, los Nombres Propios de este mes nos llevan a conocer a Rebecca Allen, una artista norteamericana cuyos trabajos se centran en el cuerpo humano mayoritariamente.

Y pasando a nuestro CD-ROM, como es costumbre, ofrecemos lo mejor que podéis encontrar en el mercado, con las mejores Demos, como la versión Trial por 30 días de Serif Page Plus, el programa de autoedición de SPC. Otra parte importante que hemos incluido es la Demo totalmente operativa, con una pequeña limitación, de Animatek World Builder, el estupendo creador de terrenos de los creadores del BonesPro, que permite generar mundos de increíble realismo, como se puede ver en las imágenes y vídeos que incluye como ejemplo. Y de la mano de REM Infográfica, una nueva demo operativa de JetaReyes pero, esta vez, para la versión 2 de 3D MAX. Incluimos también una versión de evaluación de Organica, la herramienta de diseño de Impulse, creadores del conocido Imagine, y una nueva autodemo de Autodesk, correspondiente a su herramienta AutoCAD Map. En el apartado de objetos incluimos modelos en formato 3DS y VRML, y añadimos también más texturas y Plug-Ins a nuestra colección, con un directorio dedicado especialmente a Imagine repleto de efectos, texturas, objetos y demás software de utilidad para este modelador. Y cómo no, nuestras habituales secciones de Imágenes de lectores y ejemplos de los artículos.

Por este mes, nada más, esperamos que disfrutéis con este número, y os recordamos que os estaremos esperando una vez más en el quiosco dentro de 30 días. Hasta el mes que viene.





3D WORLD
AÑO 2
NÚMERO 15

6 NOTICIAS
Espacio destinado a informar acerca de las últimas noticias acaecidas en el mundo de las 3D.

12 REPORTAJE
Como ya avanzamos en el número anterior, vamos a explicar paso a paso las técnicas y trucos empledos en la creación de personajes de síntesis, que ya han creado un nuevo género cinematográfico.

24 SOFTWARE
Este mes la sección está completamente dedicada a Smacker, una nueva y revolucionaria tecnología de compresión de vídeo que seguro plantará cara a los estándares AVI, Quicktime o MPEG.

28 HARDWARE
Comentamos unos interesantes productos para los usuarios de Mac, y dedicamos la mayor parte de la sección al nuevo reproductor DVD de Creative.

32 MODELADO ORGÁNICO
¿Por qué es tan difícil dar apariencia orgánica a un modelo? Hay varias razones, y en este artículo trataremos de dar respuestas a todas ellas.

36 COMPARATIVA
Si somos uno de los pocos privilegiados que pensamos adquirir un portátil Workstation y no sabemos por dónde van los tiros actualmente, esta comparativa nos muestra dos titánicos que podrían ridiculizar a muchos ordenadores de sobremesa actuales.

38 CLAVES DE LA INFOGRAFÍA
Este mes vamos a hablar sobre la animación antropomórfica, veremos cómo animar personajes de forma estática y personajes maleables, y terminaremos con algunos consejos y trucos muy interesantes para resolver algunas situaciones delicadas.

42 CURSO 3D STUDIO
El proceso de Render genera la imagen o animación final. Su duración suele ser directamente proporcional a la complejidad de la escena, y la representación final que se obtiene es configurable por el usuario mediante una serie de parámetros. Vamos a adentrarnos en sus secretos.

46 MAYA, LA NUEVA GENERACIÓN
El pasado 5 de marzo tuvo lugar la presentación en Madrid de Maya, la última estrella de Alias|Wavefront. 3D WORLD estuvo allí y ahora os cuenta todo lo acontecido en el evento.

52 POV-RAY
POV-Team incorporó en la última versión de POV, un conjunto de nuevas directivas en el lenguaje escénico del programa: las directivas condicionales y las funciones sobre objetos. Operaciones de gran utilidad para generar trayectorias o para construir figuras a partir de ecuaciones paramétricas.

56 CALIGARI TRUESPACE
En el presente artículo se comenzarán a mostrar las funciones principales de animación que nos permitirán realizar la secuencia de la pelea de motos de luz de la película.



Este mes, en 3D Studio, vamos a adentrarnos en los secretos que encierra el proceso de render, el más importante de todos. (pág. 42)



En el Stage Editor de Imagine se pueden crear ciertas animaciones, pero para secuencias más complejas será necesario utilizar el Action Editor. (pág. 74)

60 GRAFISMO PARA VIDEOJUEGOS

A partir del presente número, esta sección va a dar un giro de 360º para dedicarse a los videojuegos para grafistas, en lugar del grafismo para videojuegos, ya que vamos a comenzar un curso de programación en DIV.

64 WORKSHOP PROGRAMACIÓN

Para situaciones en las que no basta con la pila de matrices, OpenGL dispone de otras herramientas y, en particular, de las llamadas *Display List* que serán objeto de análisis en la presente entrega.

66 CURSO LIGHTWAVE

Es extremadamente importante poder manejar todas las herramientas de manera apropiada y cómoda, pues todas ellas forman parte del control del entorno de trabajo.

70 CURSO REAL 3D

En este capítulo se adentrará un poco más en el interminable mundo del modelado mediante curvas y superficies B-Spline, desarrollando nuevas técnicas y metodologías de modelado básico mediante el uso de objetos de apoyo adicional.

74 CURSO IMAGINE

En el *Stage Editor* se pueden cargar y colocar adecuadamente los objetos; los situamos ante la cámara, los iluminamos y preparamos la escena para la animación.

78 CURSO ALIAS POWER ANIMATOR

Este mes vamos a ver la animación a través de claves de animación o *KeyFrames*, con lo que nos meteremos a profundizar en la animación básica con Power Animator.

80 CURSO STRATA STUDIO PRO

Vamos a terminar de explicar las últimas funciones de Strata Studio Pro 1.75, por lo que finalizaremos con las funciones avanzadas de animación.

82 TÉCNICAS AVANZADAS

En línea con el artículo anterior, ofrecemos nuevos efectos aplicables al mismo ejemplo. Dichos toques harán de nuestro desarrollo una obra con mayor realismo y convicción.

86 TRUCOS PHOTOSHOP

Nos hemos ido de vacaciones y hemos tirado un montón de fotos de monumentos, plazas..., pero se nos olvidó hacer una de recuerdo. Vamos a intentar arreglar el descuido.

88 NOMBRES PROPIOS

El trabajo de Rebecca Allen se centra principalmente en la representación sintética del cuerpo como forma de comunicación y expresión creativa, y utiliza la tecnología interactiva como una extensión del cuerpo y la mente.

90 CURSO SOFTIMAGE

Una de las partes más interesantes y completas de Softimage 3D es el módulo de animación que permite recrear cualquier tipo de movimiento por complejo que sea gracias a sus avanzadas y potentes herramientas.

REFERENCIAS TÉCNICAS

Action Window. Alias Power Animator. Página 79.
Attribute Editor. Maya. Página 50.
AutoKeyframe. Alias Power Animator. Página 78.
Autoshave. Caligari trueSpace. Página 56.
Build. Real 3D. Página 71.
Constraint. Alias Power Animator. Página 79.
Dim Background. 3D Studio. Página 43.
Distance Cue. 3D Studio. Página 43.
Drag. Real 3D. Página 71.
Exaggeration. Caligari trueSpace. Página 57.
Fit All. Lightwave. Página 67.
Fit Sel. Lightwave. Página 67.
Flexors. Maya. Página 48.
Floats. POV-Ray. Página 54.
Fog Background. 3D Studio. Página 43.
Grid Units. Lightwave. Página 69.
HyperGraph. Maya. Página 47.
Image File Comments. 3D Studio. Página 44.
Lattice Box. Maya. Página 48.
Layer. Imagine. Página 77.
Layered Fog. 3D Studio. Página 44.
Length. Lightwave. Página 66.
Marking Menus. Maya. Página 46.
Maya Embedded Language. Maya. Página 46.
Measure. Lightwave. Página 66.
MeshSmooth. Modelado Orgánico. Página 33.
Motion. Softimage. Página 90.
MoveKnotPoint. Real 3D. Página 70.
OpenGL Smooth Shaded. Lightwave. Página 68.
Param Control. Alias Power Animator. Página 78.
Patch Polygon. Lightwave. Página 68.
Radiosity. 3D Studio. Página 42.
Render Blow Up. 3D Studio. Página 42.
Soft Bodies. Maya. Página 50.
Surface Tools. Modelado Orgánico. Página 33.
Switching. Maya. Página 47.
Tangent Type. Alias Power Animator. Página 79.
TglShade. Maya. Página 47.
Time Slider. Alias Power Animator. Página 78.
Vrotate. POV-Ray. Página 54.
Walk Window. Maya. Página 48.

EN EL CD-ROM...

Este mes, en 3D WORLD tenemos el mejor CD-ROM que se puede encontrar, un compacto repleto de demos, Shareware, objetos, plug-ins, efectos y demás software de utilidad para los amantes del 3D y la infografía. Nuestro recorrido comienza por los mundos virtuales que podremos crear con Animatek World Builder, uno de los mejores generadores de terrenos y paisajes disponibles en el mercado, con el cual daremos una nueva dimensión a nuestras escenas. Asimismo, incluimos una versión de prueba totalmente operativa de JetaReyes, el editor de gestos para 3D MAX que ya ofrecimos en su día para la versión 1.2 de MAX, y que esta vez regalamos preparada para funcionar en la versión 2 del modelador de Kinetix. También hemos incluido una versión Trial por un periodo de 30 días de Serif PagePlus, un sencillo programa de autoedición desarrollado por SPC, y que cuenta con avanzadas funcionalidades de autoedición, y una nueva Demo de Impulse, los creadores de Imagine, que ahora nos presentan su nueva estrella: Organica. Y como siempre, nuestras recopilaciones de Plug-ins, objetos, IPAS, utilidades, ejemplos de los artículos de 3D WORLD y creaciones de los lectores, con un apartado especial dedicado al programa de modelado y animación Imagine, para el cual ofrecemos un directorio cargado de objetos, texturas, efectos y más utilidades para trabajar con este estupendo software. En definitiva, un CD-ROM en la línea de 3D WORLD, siempre ofreciendo lo mejor que se puede encontrar.

Página 97

Serif PagePlus 5 Professional Edition



Software Publishing Corporation Holdings Inc, el conocido fabricante de software gráfico cuyo buque insignia es el galardonado Harvard Graphics, lanza ahora en el mercado español su programa "Serif PagePlus 5 Professional Edition" para Windows 95 o Windows NT 4.0, que lleva incorporadas todas las funciones de Serif PagePlus y además tiene muchas funciones nuevas, incluso potentes capacidades para editar en Internet, asistentes de diseño adicionales y optimización fotográfica.

Las nuevas capacidades de edición web de Serif PagePlus 5 facilitan al usuario la creación y edición de páginas web para Internet. Hay incluso un "comprobador de diseño de página" que ofrece consejo para evitar posibles problemas antes de editar en Internet.

El programa incluye más de 300 asistentes de diseños. Éstos permiten a los usuarios noveles crear gráficos de calidad professional con sólo contestar a unas simples preguntas. Después, el asistente se pone a trabajar creando un documento con la información facilitada. Otra función interesante es el optimizador fotográfico, que sirve para que los usuarios ajusten la graduación de colores bien en la escala de grises o bien en fotografías en color, escogiendo de entre las imágenes en miniatura que mejor representan las necesidades de su fotografía. Adicionalmente, PagePlus recuerda el emplazamiento donde se guarda esta imagen.

Serif PagePlus 5 lleva incorporada la tecnología de Formateo Inteligente, pendiente de patente, que escala texto para ajustar cambios en los tamaños del cuadro. Ya no hay que adivinar los tamaños de las fuentes pues con un sólo click de ratón automáticamente se revisa la tipografia.

Otra de sus nuevas funciones son el Calendar Wizard (una novedosa función que contiene una gran variedad de opciones con formas y diseños para poder escoger de entre ellos), Design Portfolio (una biblioteca en pantalla muy útil para almacenar nuestros diseños, fotos y logos favoritos para uso repetido, que son ideales para moverlos entre documentos) y una colección de imágenes prediseñadas, fuentes y fotos, en un archivo ampliado de más de 17.500 imágenes Clip-Art y fotos, junto con 400 fuentes. Con el uso del Clipart Manager los usuarios podrán mantener todo esto organizado.

Asimismo, muchas de las funciones que teníamos en PagePlus se han mejorado, por ejemplo, TablePlus vuelve con elementos de edición basados en celdas.

Serif PagePlus 5 está ya disponible en inglés a un precio de 14.900 Ptas + IVA. 3D

Para más información:
SPC Ibérica
Tel: (91) 637-48-54
Fax: (91) 710-33-27
Internet:
http://www.spco.com

IBM presenta el primer Microprocesador a más de 1.000 Mhz del mundo

Para más información:
IBM
Tel: (900) 100-400
Internet:
http://www.ibm.com
http://www.es.ibm.com

Ingenieros del centro de investigación de IBM en Austin (Estados Unidos) han presentado el primer microprocesador CMOS del mundo que puede operar a una velocidad de 1.000 millones de ciclos por segundo (1000 MHz o 1 GHz). Los procesadores de mayor velocidad que actualmente se comercializan están en el rango de los 300 MHz.

El procesador, que alcanzó en la prueba una velocidad de 1.100 MHz, ha sido diseñado por un equipo formado por quince ingenieros. El chip contiene un millón de transistores y para su desarrollo se ha utilizado la tecnología IBM CMOS 6X. La microarquitectura, los circuitos y las técnicas de comprobación que se derivan de este proyecto se aplicarán a los nuevos microprocesadores mediante el uso de la reciente tecnología CMOS 7S de chips de cobre, anunciada por IBM en octubre del pasado año.

El equipo de ingenieros que ha logrado diseñar el microprocesador destaca las siguientes innovaciones:

- Una unidad multifuncional que combina ciertas operaciones en un único circuito y un diseño innovador de la memoria caché.
- Un circuito dinámico que reduce drásticamente el número de estados a través del cual se propagan las señales.
- Métodos innovadores que reducen el periodo del ciclo del chip y superan el reto de generar y distribuir la señal de tiempo con un alto grado de precisión.
- Nuevas técnicas de comprobación que permiten al sistema de testación operar a una fracción de la velocidad del procesador.

Los chips de 1000 MHz fueron fabricados en el Centro de Tecnología Avanzada para semiconductores en East Fishkill de Nueva York y se testearon en el Centro de Investigación T.J. Watson, de la misma ciudad. 3D

Axis Communications anuncia el nuevo servidor de escáner AXIS700

Axis Communications, fabricante de periféricos para redes de trabajo y entornos Web, anuncia la disponibilidad del servidor de escáneres AXIS 700 Network Scan Server, un innovador periférico de red que, usado en combinación con un escáner ya existente, permite a los grupos de trabajo ofrecer acceso a la red con lo que hasta hoy era considerado un dispositivo de usuario individual. El AXIS 700 representa una manera sencilla y asequible en costes de integrar la tecnología existente para crear una solución común para todos dentro de un grupo de trabajo.

El AXIS 700 utiliza la tecnología ThinServer, propia de Axis, que permite acceso universal a los recursos del escáner a través de la red. Este avance tecnológico de Axis proporciona acceso a la red sin necesidad de un PC dedicado o un servidor de ficheros de red. Todas las funcionalidades de la red, incluido el servidor Web y las capacidades de gestión, están integradas dentro del propio producto. A ello se une la combinación de un servidor ligero mejorado y adaptado con un interfaz de gestión Web independiente de la plataforma y una CPU basada en tecnología RISC.

Algunas empresas que ya han probado el servidor de escáner AXIS 700 coinciden en señalar que incrementa el valor de su red al añadir inteligencia al escáner y hacerlo disponible para todo aquel que se encuentre conectado. Dichas empresas señalan, además, que esta característica ha permitido extender el uso del hardware que ya tenían y ha hecho posible que el escáner se convierta en un valioso recurso para la distribución de grupos de trabajo. Apenas presionando unos pocos botones del AXIS 700, un escáner conectado puede ser usado para distribuir información con destinos diversos. Un usuario típico tan solo tiene que poner el material en el escáner, seleccionar un destino y hacer click en "enviar". Los usuarios disponen de una completa gama de opciones de escáner a elegir en función de sus aplicaciones y de la frecuencia con que digitalicen documentos.

Asimismo, cuenta con una opción denominada *Del escáner al E-mail*, que permite enviar imágenes digitalizadas por correo electrónico. El servidor de escáner AXIS 700 permite el envío directo de información digitalizada tanto dentro de la propia organización como hacia el exterior. Para la mayoría, ésta resulta ser la opción preferida para la ditribución de documentos e imágenes, ya que permite ahorrar, por ejemplo, los costes de un envío de fax.

Esta opción tiene sus homólogas en las funciones *Del escáner al archivo* y *Del escáner al navegador*. La primera de ellas almacena los documentos digitalizados como ficheros de imagen directamente a un disco compartido en la red (se trata de una solución ideal para la automatización, gestión de documentos y tareas de Workflow). La segunda, por su parte, envía imágenes al cliente de la red mediante un navegador Web. Ésta última es una elegante opción para entornos de trabajo con un departamento técnico pequeño o inexistente, así como para usuarios que digitalizan documentos con poca frecuencia.

Una de las diferencias clave entre el servidor de escáner de red AXIS 700 y otras soluciones de digitalización compartida es la posibilidad de soportar escáneres de diferentes fabricantes. En vez de incluir o estar vinculado a un tipo de escáner, el AXIS 700 puede convertir la mayoría de los escáneres planos en dispositivos listos para ser conectados a la red. Además, gracias al soporte de alimentadores automáticos de documentos (ADFs), así como de compresión y módulos proceso de imagen (IPC), este servidor de escáner en red puede ser integrado con las principales soluciones de digitalización de fabricantes como Fujitsu, Hewlett-Packard y Canon.

El servidor AXIS 700 soporta las más conocidas aplicaciones de imagen y de reconocimiento óptico de caracteres (OCR) disponibles en el mercado. También integra fácilmente todas los paquetes actuales de *Workflow* (flujo de trabajo), *Groupware* (trabajo en grupo) y gestión de documentos, de manera que ni los usuarios finales ni los administradores de la red tienen por qué preocuparse por la cuestiones relativas a la compatibilidad. Una de las claves de la facilidad con que los usuarios pueden abrir y visualizar imágenes digitales reside en el soporte que el producto incluye, tanto para TIFF 6.0 como para el formato de intercambio de archivos JPEG (JFIF). Ambos se han convertido en estándares de facto en formatos de imagen para visualizar y manipular archivos, asegurando que los usuarios tengan la posibilidad de usar herramientas con las que ya están familiarizados.

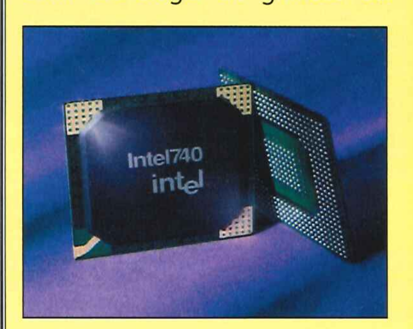
EL servidor de escáner en red AXIS 700 está ya disponible en España a un precio recomendado de 299.000 pesetas. 3D

Más información:
Axernet S.A
Tel: (91) 803-02-44/411-47-85
Fax: (91) 803-54-52 Fax. 91- 411 56 12
Internet: *http://www.axernet.es*

Afina Sistemas
Tel: (91) 411-47-85
Fax: (91) 411-56-12

Intel anuncia el acelerador de gráficos 3D Intel740

Intel Corporation anuncia su nuevo chip acelerador de gráficos, el Intel740. Optimizado para el procesador Pentium II con AGPsets de Intel, el Intel740 es la fundación de los esfuerzos realizados por Intel para mejorar la experiencia visual de los usuarios de PCs. La arquitectura Hyper Pipelined 3D y la potente aceleración de vídeo y 2D introducen un alto nivel de realismo en el PC para una informática de primera categoría, real como la vida misma. Al tiempo que la rápida adopción de la industria del interfaz AGP (*Accelerated Graphics Port*) conducida por Intel, junto con la avanzada tecnología de gráficos del Intel740, permiten una nueva generación de aplicaciones profesionales y de consumo para PC.



Según fuentes de Intel, el chip promete proporcionar altas prestaciones a los desarrolladores y usuarios finales. Esto, junto a la combinación del Intel740 y su sólida implementación en el API Direct3D estándar, significará una mayor potencia y flexibilidad para los desarrolladores de juegos y contenidos más interesantes para los usuarios finales.

El Intel740 está diseñado para equilibrar las prestaciones en el procesador Pentium II utilizando las avanzadas unidades de coma flotante de dicho procesador, así como las altas capacidades de ancho de banda y mejora de gráficos de los AGPsets. Esto proporciona una experiencia gráfica rica a un precio de volumen. Esta avanzada tecnología de gráficos es una excelente solución para títulos 3D de nueva generación como Quake II, Red Line Racer, Incoming y Tonic Trouble, entre otros.

Intel ha trabajado de forma estrecha con una amplia gama de desarrolladores de software para permitir contenidos de próxima generación para el procesador Pentium II con AGP y gráficos Intel740. Estas innovadoras aplicaciones 3D, como la visualización de datos, el CAD de consumo y los navegadores de Web en 3D, así como los títulos DVD de alta calidad y los nuevos juegos, que utilizan la potente aceleración del Intel740, contribuyen a la transformación de la informática en una experiencia visual incomparable para los usuarios de PCs corrientes.

El Intel740 estará disponible en volumen para los fabricantes gráficos de empresas terceras y OEMs, y su precio es de 34'75 dólares (unas 5.400 pts) para cantidades de 10.000 unidades. 3D

Más información:
Intel
http://www.intel.com/pressroom

Apple amplía la gama Power Macintosh G3 con un nuevo modelo para los profesionales más exigentes

Apple presenta en España un nuevo modelo de ordenador Power Macintosh G3 minitorre para clientes profesionales, caracterizado por una mayor velocidad de almacenamiento en disco y de comunicación en red, y por un superior rendimiento gráfico. Estas nuevas características de gama alta son especialmente idóneas para nuestros clientes en los mercados de edición profesional y creación de contenido que quieren complementar el alto rendimiento del procesador G3, con la mayor velocidad posible de Entrada/Salida para manejar sus enormes archivos gráficos y multimedia. Con el nuevo sistema, este tipo de clientes se beneficiarán del veloz acceso a disco que posibilita la interfaz Ultra/Wide SCSI, y de la rápida comunicación en red que brinda la Tarjeta PCI Fast Ethernet 10/100Base-T.



Otro requerimiento clave de este mercado es manejar gráficos a muy alta velocidad, exigencia que es cumplimentada con la incorporación de una tarjeta aceleradora gráfica 2D/3D de 128 bits. Esta tarjeta permite, además, la conexión al sistema de un segundo monitor de manera estándar.

La nueva configuración está basada en un procesador PowerPC G3 a 266 MHz, con 512K de Backside Caché de nivel 2. Incluye 128 MB de memoria RAM (ampliables a 384 MB), circuitería de aceleración gráfica de 64 bits en placa con el chip ATI 3D Rage II+ y 6 MB de SGRAM, unidad de disquetes y lector de CD-ROM ATAPI 24x, tarjeta PCI Fast Ethernet 10/100Base-T, tarjeta PCI Ultra/Wide SCSI (40 MB/s), disco duro Ultra/Wide SCSI de 4 GB de capacidad y aceleradora gráfica 2D/3D de 128 bits.

Desde su estreno el pasado noviembre, la línea de sistemas Power Macintosh G3 de Apple se ha convertido en el producto hardware que ha logrado un mayor éxito en su lanzamiento en la historia de Apple, con ventas superiores a las 133.000 unidades en menos de dos meses.

La nueva configuración Power Macintosh G3/266 minitorre 128MB/4GB UW-SCSI/24xCD está ya disponible a través del canal de distribución de Apple en España, y su precio orientativo de venta al público es de 729.000 Ptas más IVA. 3D

Más información:
Apple Computer Spain.
Tel: (91) 663-17-80
Fax: (91) 663-17-90
Internet: ***http://www.apple.es***

Autodesk adquiere nueva tecnología de programación

Autodesk ha anunciado la adquisición de la tecnología del lenguaje de programación Vital LISP de Basis Software. Este lenguaje se basa en la tecnología de programación de núcleo orientada a objetos ObjectARX de Autodesk. Vital LISP también abre nuevas posibilidades para el lenguaje de programación AutoLISP de Autodesk. Con ello, los usuarios de AutoCAD 14 y otros productos de Autodesk podrán conseguir con esta tecnología una mayor flexibilidad y libertad durante la personalización y adaptación del software. Esta adquisición demuestra el compromiso de Autodesk con sus usuarios y clientes, que han invertido tiempo y recursos usando la tecnología de programación AutoLISP para la personalización de AutoCAD desde 1985. El lenguaje AutoLISP es utilizado habitualmente por más de un millón de usuarios de AutoCAD del mundo para personalizar sus programas.

Por su parte, Vital LISP proporciona la tecnología de núcleo de desarrollo necesaria para el nuevo lenguaje de programación Visual LISP de Autodesk, que incorporará un nuevo conjunto de avanzadas herramientas de personalización basadas en tecnología orientada a objetos para los usuarios de AutoCAD 14. Las ventajas de la programación con Visual LISP serán la mejora de la productividad mediante el uso de una nueva herramienta con interfaz visual para una rápida y fácil personalización, permitiendo crear aplicaciones en menos tiempo; mejora de los tiempos de trabajo, gracias al aumento de la velocidad a la que se ejecuta el código LISP en el interfaz basado en ObjectARX de AutoCAD 14; mejora del interfaz con Microsoft Windows ActiveX para el Modelo de Objetos de AutoCAD, proporcionando una mayor flexibilidad en la integración entre aplicaciones, y la posibilidad de compilación de rutinas LISP mejorando su funcionamiento, la rapidez de ejecución y evitando el acceso no deseado al código fuente del software de la aplicación del usuario. 3D

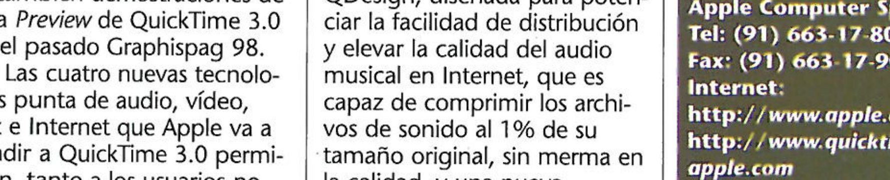
Más información:
Autodesk
http://www.autodesk.com

Apple añade a Quicktime 3.0 nuevas tecnologías punta de música, vídeo, voz e Internet

Apple ha adquirido la licencia para incorporar otras cuatro innovadoras tecnologías de música, vídeo, voz e Internet a QuickTime 3.0, la próxima versión de la conocida tecnología multimedia QuickTime, convertida hoy en uno de los estándares de la industria para crear y publicar información multimedia digital para Mac OS y Windows. La versión final de QuickTime 3.0, con todo este abanico de innovaciones, se lanzará al mercado en las próximas semanas.

Como avance, Apple ha colocado en Internet, en la dirección http://www.quicktime.apple.com/preview/ de su Web, una versión previa de QuickTime 3.0 para que cualquier usuario pueda tomar un primer contacto con esta avanzada tecnología. Apple realizará también demostraciones de esta *Preview* de QuickTime 3.0 en el pasado Graphispag 98.

Las cuatro nuevas tecnologías punta de audio, vídeo, voz e Internet que Apple va a añadir a QuickTime 3.0 permitirán, tanto a los usuarios no expertos como a los profesionales de la creación, producir excepcional contenido multimedia para Internet, de superior calidad y menor necesidad de ancho de banda. Éstas son el compresor de vídeo *Sorenson Vision*, que garantiza una sustancial mejora en calidad y nitidez para Internet. Esta tecnología escalable redefine la compresión de vídeo, proporcionando resultados sorprendentes tanto al publicar en CD-ROM como al hacerlo vía Internet para modems de incluso 28.8 kbps; PureVoice de QUALCOMM, una tecnología aplicable a una amplia variedad de aplicaciones software que utilizan voz, que comprime la voz a un formato digital extraordinariamente compacto, manteniendo una reproducción de calidad para el oyente; tecnología de compresión de audio digital de QDesign, diseñada para potenciar la facilidad de distribución y elevar la calidad del audio musical en Internet, que es capaz de comprimir los archivos de sonido al 1% de su tamaño original, sin merma en la calidad, y una nueva Biblioteca de Instrumentos de Roland Corporation, patrones de alta calidad que, gracias a QuickTime, pueden aplicarse a la reproducción MIDI.



QuickTime 3.0 incluirá, además, nuevos servicios para Internet, una nueva arquitectura de efectos especiales y nuevas características de vídeo profesional. Con QuickTime 3.0 todas las capacidades de creación, edición y publicación multimedia de QuickTime, estarán disponibles por vez primera para plataformas Mac OS, Windows 95 y Windows NT. Esta compatibilidad multiplataforma proporcionará un nivel de potencia y flexibilidad sin precedentes para los usuarios de QuickTime en cualquiera de las principales plataformas de ordenador. 3D

Más información en:
Apple Computer Spain
Tel: (91) 663-17-80
Fax: (91) 663-17-90
Internet:
http://www.apple.es
http://www.quicktime.apple.com

Micrografx lanza picture Publisher 8.0

Micrografx Inc, el conocido fabricante de software gráfico, anunció el pasado mes la disponibilidad inmediata de Picture Publisher 8, su conocida aplicación de edición de imágenes para gráficos en Internet que incluye efectos visuales patentados, cientos de filtros y la curva más baja de aprendizaje entre las herramientas de nivel profesional.

Creado sobre las bases de las herramientas de diseño gráfico y de imágenes de Micrografx, las nuevas funciones de Picture Publisher 8 fortalecen tanto a los profesionales como a los entusiastas de Internet al facilitar la producción rápida y sencilla de imágenes de alta resolución sin comprometer la calidad o la rapidez a la hora de cargar la página.

Picture Publisher 8, con su amplia colección de herramientas de edición de imágenes de alta resolución, ofrece la amplitud y profundidad de funciones que permite a los profesionales de Internet obtener los resultados deseados rápidamente. Con un completo conjunto de herramientas y contenido para retoque fotográfico, creación de imágenes y pintura digital, gráficos de Internet y mucho más, Picture Publisher se convierte en una herramienta muy a tener en cuenta.

El programa ayuda a los usuarios a aprovechar al máximo su potencia en la edición de imágenes de alta resolución, a la vez que es una de las aplicaciones de la industria más fácil de usar. Asimismo, los profesionales del diseño en Internet se beneficiarán del nuevo nivel de potencia y productividad que les aporta Picture Publisher 8 con sus funciones para crear imágenes y efectos web impactantes y eficaces. Herramientas de edición globalizadoras, *Asistentes* de contenido interactivo Web y efectos únicos que permiten la creación del más variado y atractivo contenido profesional Web, a la vez que la amplia reducción del color y herramientas de edición de paletas permiten cargar rápidamente imágenes web eficaces que mantienen brillantes colores.

Picture Publisher 8 ofrece, además, un completo soporte a los formatos de archivos más conocidos, incluidos GIF transparentes y entrelazados, *rendering* progresivo JPG, PNG y FlashPix. Sus avanzadas opciones de archivo y previsualizaciones interactivas del tiempo de bajada, compresión y transparencia ayudan a asegurar la creación de contenido de alta calidad para ser usado en Internet. Incluso es posible crear archivos GIF animados rápida y fácilmente gracias al apoyo perfectamente integrado a la animación GIF, que saca todos los beneficios de las herramientas y efectos de Picture Publisher 8. A nivel modular, Picture Publisher 8 ofrece apoyo total a plug-ins post-venta, pero no depende de ellos, lo que permite a



los usuarios de Picture Publisher un uso más eficaz de su tiempo y dinero.

Además Picture Publisher 8 aporta la solución más completa de edición de imágenes con una robusta gama de herramientas de alta resolución para proporcionar a los profesionales gráficos la rapidez, precisión y calidad de producir velozmente los resultados deseados. Con cientos de herramientas para la edición de fotos, creación de imágenes, pintura digital, gráficos Web y muchos más efectos, Picture Publisher está preparado para cualquier proyecto imaginable.

Picture Publisher 8 se encuentra ya disponible para ordenadores que utilizan Windows 95 y Windows NT a un precio de venta al público orientativo de 15.900 Pts.+ IVA, o al precio de actualización desde versiones anteriores de 13.900 Ptas + IVA. 3D

Para más información:
Micrografx Ibérica
Tel: (91) 710-35-82
Fax: (91) 710-33-27
Internet:
http://www.micrografx.com

Microsoft anuncia la beta de NetShow 3.0

Microsoft Ibérica anunció recientemente la versión beta de Microsoft NetShow 3.0. Esta nueva versión ofrecerá a las compañías las herramientas necesarias para trabajar con sonido y vídeo sobre Internet e Intranets corporativas. Asimismo, Microsoft NetShow 3.0 incluye Windows Media Player, lo que permitirá que más usuarios puedan disfrutar de una amplia oferta sin tener que trabajar con diferentes formatos de ficheros. Con esta versión beta, Microsoft intenta ofrecer una alta calidad de vídeo y sonido además de utilizar los estándares clave, como el ASF (*Advanced Streaming Format*), para desarrolladores y proveedores de contenido.

Microsoft ha trabajado con más de 50 compañías de hardware y software como Intel, Adobe, RealNetworks o Vivo para establecer ASF como uno de los formatos de archivo estándar para multimedia. Al estandarizar sobre ASF, las compañías podrán tener la seguridad de que cualquier persona pueda visualizar contenido de vídeo y audio, sin importar quién es el autor. Asimismo, esta estandarización supone que las empresas puedan elegir y centralizar su desarrollo sobre un único formato, lo que permitirá tener herramientas de alta calidad para la creación de contenidos.

Windows Media Player, incluido en Microsoft Netshow 3.0, ejecutará no sólo el formato ASF, sino también RealAudio/RealVideo, audio y vídeo de Windows y contenido QuickTime, entre otros. Asimismo, NetShow 3.0 emplea la tecnología de gama líder denominada *Intelligent Streaming*. Esta tecnología, combinada con los últimos avances en compresión de vídeo, aumenta significativamente la calidad de vídeo que se envía a través de Internet. Con NetShow 3.0 se podrá ver vídeo a pantalla completa con imágenes uniformes y de mejor calidad e intensidad.

Esta nueva versión está altamente integrada con otros productos de Microsoft, pudiendo ser compatible con una amplia gama de aplicaciones, como los servicios de noticias que ofrecen información selectiva a las empresas que dan formación a sus empleados en localizaciones remotas.

NetShow 3.0 trabaja con Microsoft Windows NT Server, lo que permite a los administradores tener a su disposición una amplia gama de herramientas de seguridad para controlar qué contenidos pueden ser visualizados y quién va a tener acceso a ellos. 3D

Más información
Microsoft
http://www.microsoft.com/spain

Microstation Triforma, disponible en castellano

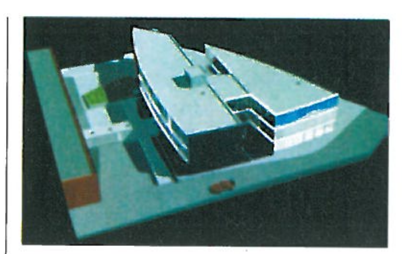
Bentley Systems Ibérica acaba de anunciar el lanzamiento en castellado de Triforma, su producto orientado al diseño arquitectónico y la documentación de la construcción. Su sencilla utilización y su alto rendimiento le convierten en uno de los mejores productos para los usuarios de A/E/C para combinar el diseño básico y el modelado conceptual arquitectónico en 3D.

Triforma, con su proceso de "modelo de construcción único", representa un avance en relación al procedimiento CAD tradicional para el diseño de producción y la construcción de maquetas virtuales, ya que se centra en la creación de un modelo 3D que abarca el proceso completo de producción arquitectónica.

La modelización conceptual que proporciona el "modelo de construcción", antes mencionado permite crear, evaluar y modificar de forma rápida los conceptos de diseño, trabajando en tres dimensiones con herramientas para esculpir, modificar, cortar y visualizar el diseño conceptual. Además, el paquete incluye herramientas para crear imágenes de gran realismo y elaborar animaciones, posibilitando la asignación de patrones y texturas al modelo.

Otra de las ventajas del modelo de construcción es que todos los gráficos 2D, incluyendo plantas, perfiles y alzados, se generan a partir del modelo 3D, y cualquier cambio en el mismo se verá reflejado automáticamente en los documentos pertinentes. Este método reduce errores y produce un gran ahorro de tiempo.

Las funcionalidades de Triforma en lo referente a informes son potentes y rápidas. El informe detallado de la lista de materiales que da la cantidad y coste de los componentes del modelo, y el procedimiento *"what-if?"* para el



control presupuestario, son algunas de ellas. Asimismo, es compatible con los sistemas operativos DOS, Windows, UNIX y Macintosh. En línea con los otros productos, de Bentley, Triforma ofrece un interesante entorno de programación. Las aplicaciones se desarrollan con el lenguaje de programación de MicroStation (MDL) y con el sencillo MicroStation BASIC. 3D

Para más información:
Bentley Systems Ibérica
Tel: (91) 372-84-94
Fax: (91) 307-62-85
Internet:
http://www.bentley.com

Nueva versión 3.0 en castellano de Art•lantis

NMI programación, distribuidor oficial en España de Archicad, ha anunciado la disponibilidad en español de la versión 3.0 de Art•lantis, un programa de cálculo fotorrealista en 3D directamente conectado con ArchiCAD y ZOOM. Art•lantis permite generar las vistas fotorrealistas, animaciones y entornos de Realidad Virtual de alta calidad muy rápidamente y con todo tipo de efectos como sombras, reflejos, transparencias, relieves, etc.

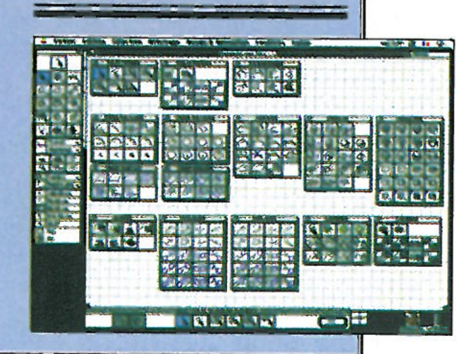
Art•lantis permite cambiar la imagen con sólo arrastrar la textura deseada sobre la superficie a la que se desea aplicar. Trabaja indistintamente con texturas vectoriales denominadas *Shaders* o con mapas de bits, y una ventana de previsualización refleja los cambios en tiempo real.

Dado que el entorno de trabajo es tridimensional, se puede cambiar la posición de la cámara para obtener varias vistas del entorno 3D. Las luces pueden proceder de una lámpara, siendo así de proyección radial; del sol siendo en este caso de proyección paralela; o de un foco, obteniendo una proyección focal y se puede cambiar su posición, su dirección o cualquiera de sus parámetros de alcance, intensidad, etc. Además, se pueden incluir efectos como niebla en la visualización.

Art•lantis permite también crear animaciones y moverse por dentro y fuera del entorno 3D creado por el usuario. Al trazar el recorrido de la cámara, la pantalla de previsualización muestra en tiempo real la animación e incluso se modifica la trayectoria a medida que se va moviendo. El fondo de la vista puede ser desde un color simple o degradado hasta una imagen tridimensional envolvente, lo que añade aún más realismo a las animaciones.

Además, con Art•lantis también se pueden crear entornos de Realidad Virtual en los que los clientes se pueden mover a su antojo por dentro y fuera del diseño tridimensional, con sólo mover el ratón. Art•lantis está disponible para entornos Windows 95, Windows NT, Apple Macintosh y Power Macintosh. 3D

Para más información:
NMI Programación
Tel: (91) 413-82-32
Fax: (91) 413-82-21



Personajes de síntesis (II)

Dentro de algunos meses llegará a nuestras pantallas la nueva producción de Pixar "A bug's life", segunda película diriga por John Lasseter y realizada por completo con técnicas de animación por ordenador.

Como ya avanzamos en el número anterior de 3D World, durante los próximos meses vamos a explicar, paso a paso, las técnicas y trucos empledos en la creación de estos personajes de síntesis, que ya han creado de hecho un nuevo género cinematográfico.

Con las indicaciones que dimos en la primera parte, ya tendría que estar preparado nuestro ordenador en casa, con el software necesario para seguir las explicaciones y ejercicios que iremos viendo. En general, vamos a usar la versión 1.2 de 3D Studio MAX en castellano, y más adelante dos estupendos módulos para este programa de la empresa REM Infográfica (MetaReyes y ClothReyes), así como el módulo de animación de personajes Character Studio de la propia Kinetix.

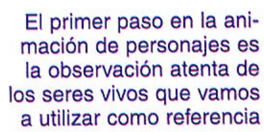

El primer paso en la animación de personajes es la observación atenta de los seres vivos que vamos a utilizar como referencia

Figura 1. Para animar un conjunto de objetos, primero debemos establecer una jerarquía entre ellos, de forma que identifiquemos claramente qué objetos arrastran a otros en su movimiento.

3D World ya ha suministrado versiones de demostración de estos programas en los discos de portada de números anteriores, por lo que los lectores habituales no deben tener ningún problema. Si en el caso particular de cada lector, preferimos usar otro software como Lightwave, Caligari o Softimage, tampoco hay de que preocuparse, ya que las explicaciones siempre serán lo suficientemente amplias como para que las técnicas que vamos a explicar aquí se puedan usar en otros programas sin dificultad.

Softimage, por ejemplo, está mucho mejor dotado que 3D Studio MAX para cualquier tipo de animación (también es cierto que cuesta cinco veces más), y en su momento daremos algunos ejemplos concretos de cómo



usar las herramientas de ésta y otras de modelado y animación en 3D.

Esqueletos

Este primer capítulo va a ser un poco "rollo", al menos desde el punto de vista de que no vamos a crear ningún personaje, ni vamos a modelar un teléfono que baile como Fred Astaire o Nureyev. Lo que vamos a hacer es tratar de comprender y aprender a utilizar la animación mediante huesos o *bones*.

La base de la animacion de personajes, tal y como se entiende hoy día, reside en la aplicacion de reglas de cinemática inversa a esqueletos de referencia. ¡Guau, esto suena a algo súper-técnico y de última hornada tecnológica! En realidad es algo bastante simple.

Pensemos por un momento en la última vez que fuimos a un parque de atracciones y montamos en el Túnel del Terror, o alguno de esos recorridos en coches eléctricos en el que parece que nos puede caer cualquier cosa en la cabeza menos un saco de billetes. Normalmente, se trata de disfraces que recuerdan a personajes famosos como Drácula o el hombre lobo. Casi todo el mundo es consciente de que se trata de un muñeco vestido y maquillado, que la mayoría de las veces se reduce a un armazón de metal revestido de gomaespuma o algo parecido.

Estudio del esqueleto

Lo que vemos, pues, es algo que está montado sobre un *esqueleto* de apoyo. El esqueleto puede estar hecho de cualquier material y no tiene por qué reproducir con exactitud el esqueleto original del personaje que representa. De hecho, suele suceder todo lo contrario, y estos muñecos consisten en poco más que dos barras de metal soldadas.

Cuando tratamos de reproducir el movimiento de un ser vivo, lo que tenemos que hacer es tratar de observar cuáles son los elementos

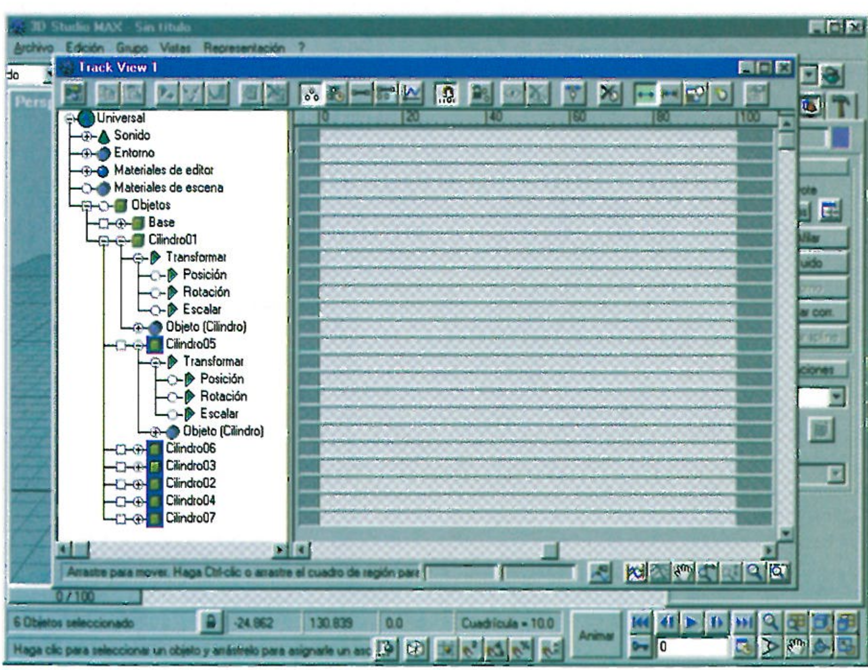


Figura 2. En 3D Studio MAX podemos comprobar las vinculaciones establecidas mediante esta ventana.

clave de ese movimiento. Por ejemplo, si queremos que nuestro muñeco de feria alce un hacha amenazadoramente sobre los visitantes, basta con reproducir el giro vertical del hombro o del codo. No es necesario que recreemos todo el movimiento rotatorio de la cabeza del húmero (el primer hueso del brazo) o las complejidades de la mano.

Por tanto, el primer paso en la animación de personajes es la observación atenta de los seres vivos que vamos a utilizar como referencia. Esto puede evitarnos mucho trabajo porque, aunque siempre es mejor tener un esqueleto que reproduzca el original lo más fielmente posible, puede ser complicadísimo, e incluso exceder la capacidad de cálculo de nuestro ordenador.

Hay que fijarse en que no es imprescindible que el personaje de estudio sea idéntico al que vamos a animar. Es decir, cuando los animadores de Disney crearon la alfombra mágica de Aladdin (un personaje fantástico en todos los sentidos), ¡no se fijaron en una alfombra! Lo que hicieron fue estudiar la gesticulación más normal de una persona en ciertas situaciones: miedo, ataque, sueño, etc. Así, al principio de la película vemos una alfombra que no habla, pero que expresa perfectamente su desconfianza hacia los dos personajes extraños mediante el movimiento de los flecos superiores.

Si lo que pretendemos es conseguir que un elefante se mueva como un perro (George de la Jungla), debemos estudiar los movimientos y limitaciones del perro y aplicarlos al esqueleto del elefante.

Recordemos que en el artículo anterior habíamos recomendado releer los textos del colegio, o los que hubiera por la biblioteca de casa, sobre anatomía, tanto de los seres humanos como de otros animales. Es muy importate estudiar y comprender no sólo que este hueso se llama de una forma u otra, sino que además se mueve en este o en ese sentido y, sobre todo, con ciertas limitaciones de giro.

De todo esto sacamos que el movimiento de los personajes de síntesis no se obtiene animando directamente las mallas que los forman, sino un esqueleto de referencia al que están vinculadas. Este esqueleto suele ser invisible y se reduce a unos pocos polígonos en los que destaca una zona de cabeza y otra de cola.

El movimiento de los personajes de síntesis no se obtiene animando directamente las mallas que los forman, sino un esqueleto de referencia al que están vinculadas

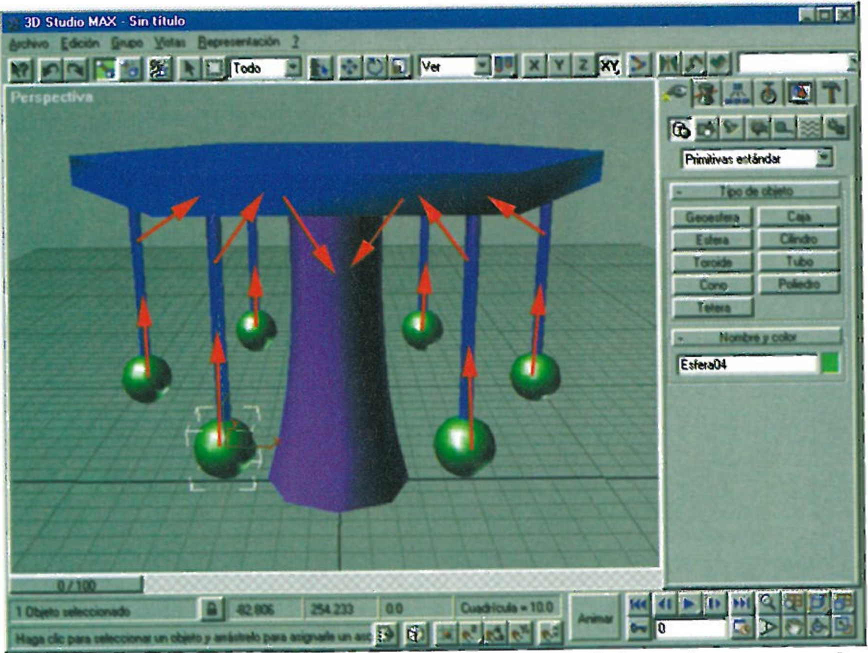


Figura 3. Si añadimos un elemento más en cada una de las cadenas que cuelgan de la plataforma, se hace más evidente la relación entre padres e hijos.

Como ocurre con los esquelos animales, los huesos en 3D se pueden agrupar en grupos y jerárquicamente; esto quiere decir que si en el ser humano los huesos de la mano forman un grupo separado y diferenciado de las costillas, lo mismo puede ocurrir con los de síntesis.

Jerarquías

Hay que detenerse aquí un momento para comprender bien lo que significan estas propiedades. Casi seguro que todos los lectores han montado alguna vez una maqueta o un puzzle; si no ha sido así es que se han olvidado de vivir unos cuantos años de infancia. Cuando comenzamos a colocar las piezas de un puzzle no lo hacemos por las buenas (bueno, conozco a un par que sí lo hacen así), sino que *agrupamos* aquellas que comparten alguna característica como su color o la posición, sobre todo las que forman el borde, para tener pistas que reduzcan el número de posibilidades y faciliten el montaje.

De la misma forma, cuando nos ponemos a modelar un personaje animado hay que agrupar sus componentes para que no se pierdan en la progresiva complejidad de las proyecciones alámbricas. A este respecto hay algunas reglas que todo modelador debe poner en práctica, sobre todo a la hora de etiquetar los objetos que hay en escena.

Además, en el esqueleto de un animal hay una cierta *jerarquía*, de forma que unos huesos arrastran a otros en su movimiento. Un dedo de la mano puede moverse libremente sin que eso suponga oscilación alguna del hombro, pero es imposible mover el brazo sin trasladar consigo todos los dedos de la mano.

En los programas de 3D, como 3DS MAX, podemos

Cuando nos ponemos a modelar un personaje animado hay que agrupar sus componentes para que no se pierdan en la progresiva complejidad de las proyecciones alámbricas

Huesos y esqueletos

Cada programa tiene una forma particular de aplicar lo que hemos aprendido en este artículo, pero los principios son exactamente los mismos. Siempre habrá que definir una jerarquía entre padres e hijos con las limitaciones que hemos citado: un padre puede tener muchos hijos, pero cada hijo sólo puede tener un padre. Al mismo tiempo, un mismo objeto puede ser padre e hijo a la vez.

Igualmente, siempre habrá que definir las limitaciones de movimiento cuando apliquemos cinemática inversa, aunque la forma de hacerlo será particular de cada programa.

Hasta ahora hemos trabajado sobre una premisa falsa en el mundo de la animación, y es que hemos trabajado directamente sobre los objetos de la escena. Con este método podemos animar sistemas mecánicos sencillos, pero no seres humanoides ni otras criaturas, pues no hay forma de hacer que los miembros se retuerzan, estiren y contraigan según la posición que adopten.

En la siguiente entrega veremos cómo se aplica lo que hemos visto aquí a un artificio denominado *huesos*, que sirven de verdadero esqueleto a los personajes de síntesis. Veremos cómo construir un esqueleto completo a partir de dibujos o fotografías, estableciendo las jerarquías necesarias entre sus todos sus elementos. También empezaremos a trabajar con el módulo Character Studio de Kinetix, que ahorra una gran catidad de trabajo en la definición de los grados de libertad de la cinemática inversa.

Mientras tanto, cualquier ejercicio que se realice es bueno. Hay que perder el miedo a modelar la primera tontería que se nos ocurra y tratar de animarla según las reglas que hemos visto en el artículo. Puede que parezca un consejo sin sentido, pero no grabe el trabajo diario. Hay que procurar partir siempre de cero para acostumbrarnos a recorrer todos los pasos sin olvidarnos de ninguno.

Sobre todo, es importante que sigamos observando las cosas que nos rodean, los animales domésticos, documentales de vida salvaje y nuestros propios andares cuando nos dirigimos al trabajo o a casa de un conocido. Sólo de este análisis podrá salir una buena animación.

establecer el mismo tipo de vinculaciones que existen en nuestro cuerpo. Entre dos objetos (ya sean sillas, hélices o huesos) sólo puede establecerse un tipo de relación: padre/hijo. La diferencia entre el padre y el hijo es que un objeto padre puede tener muchos hijos, pero un hijo sólo puede tener un padre. Ésta es una regla de animación fácil de recordar, pues hay una equivalencia con las relaciones familiares reales. Veamos un ejemplo sencillo.

En la segunda pantalla que ilustra este artículo vemos una escena bastante simple: un eje central sobre el que se sitúa una plataforma, de la que a su vez cuelgan varios objetos alargados, imitando la estructura de un tiovivo. El soporte central es el *padre* de todos los objetos que hay en la escena, pero para establecer esta relación no hay que vincular cada uno de los objetos restantes con este soporte, sino establecer la jerarquía adecuada.

Lo que hacemos es activar la herramienta de vinculación, situada en la parte izquierda de la barra de herramientas superior (está resaltado en rojo en la imagen). El puntero parece igual, pero si lo acercamos a la plataforma superior en cualquiera de las vistas, veremos que cambia de forma, apareciendo dos cubos en los extremos de una cadena. Si pulsamos el botón izquierdo del ratón sobre el hexágono que representa la plataforma y arrastramos el ratón hasta el poste central, veremos que uno de los cubos se vuelve de color verde, al tiempo que aparece una línea punteada desde el centro geométrico del primer objeto.

Soltamos el botón izquierdo al llegar al poste y repetimos la operación con cada uno de los seis cilindros que cuelgan de la plataforma, pero en esta ocasión, el objeto con el que los vamos a vincular no es el poste, sino la plataforma superior. ¿Qué es lo que hemos conseguido? Si abrimos el panel de jerarquías lo veremos en seguida (en la barra superior de iconos, hacia la derecha).

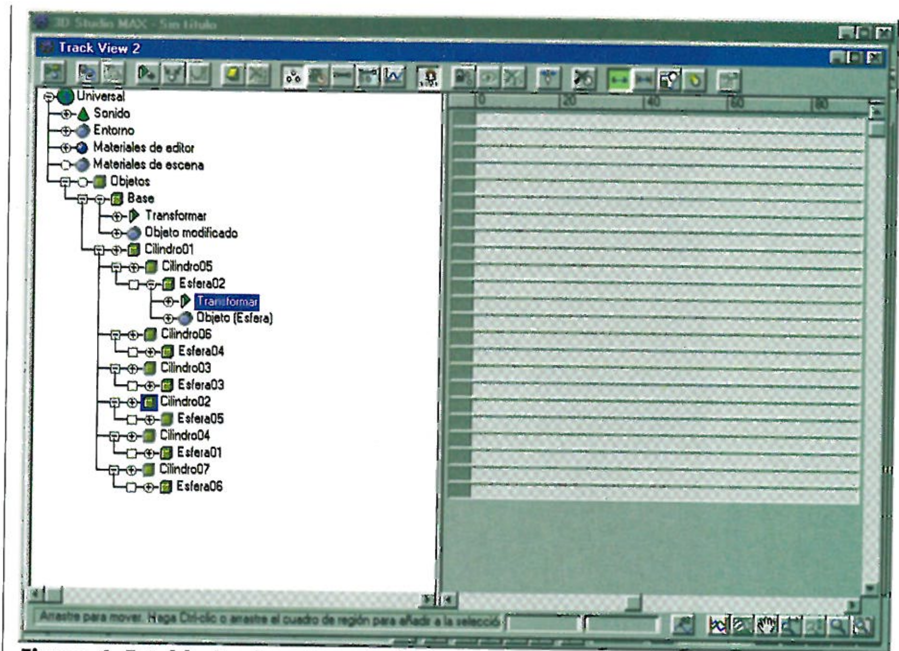


Figura 4. Estableciendo estas relaciones podemos aplicar el método de animación mediante cinemáticas directas: los padres arrastran a los hijos en todos sus movimientos.

Como podemos apreciar en la siguiente ilustración, hay varios objetos en la escena, pero no todos ellos cuelgan directamente de la primera rama del árbol, sino que sólo el que etiquetamos como *base* lo hace. Por debajo de la base está la plataforma (*cilindro 1*) y, a su vez, el resto de objetos (*cilindros 2 a 7*) cuelgan de ésta. La base es el *padre* de la plataforma, y a su vez ésta es padre del resto de los objetos. Se mantiene la norma que habíamos establecido: un padre puede tener varios hijos, pero cada hijo sólo puede tener un padre.

Siguiendo con el equivalente real, un hijo puede ser, a su vez, padre de otros hijos, de forma que la jerarquía entre los objetos de una escena puede ser tan larga y compleja como nosotros queramos y necesitemos. En la práctica, sin embargo, no es habitual que estas relaciones se prologuen demasiado.

Cinemática directa

Antes de explicar lo que es la cinemática inversa, vamos a ver lo que es la cinemática directa. Aunque el movimiento general de los personajes se consigue con la segunda, la cinemática directa no es menos importante, y veremos la necesidad de dominarla cuando tengamos que animar el movimiento de una cabeza cuando el resto del cuerpo vaya corriendo de un sitio a otro de la escena.

Si hemos entendido el concepto de jerarquía y las relaciones entre padres e hijos esto tiene que ser muy fácil. Volvamos al ejemplo del tiovivo y añadamos una esfera en el extremo de cada uno de los cilindros que cuelgan de la plataforma. Ahora vamos a establecer un vínculo entre cada una de estas esferas y el cilindro del que cuelgan; cada esfera es hijo de su cilindro correspondiente que, a su vez, es hijo de la plataforma, que, a su vez, es hijo de la base central.

¿Qué pasaba cuando éramos pequeños y nuestros padres decidían que había que ir de vacaciones a tal sitio? Que por mucho que nos desagradara les seguíamos a todas partes. Esta es una relación *directa* entre el movimiento de padres e hijos, y en el universo 3D que creamos en los programas de modelado y animación pasa exactamente lo mismo: los hijos siguen a los padres (aunque aquí los padres tienen la ventaja de que los hijos no piden que se pare el coche para hacer sus necesidades por el camino).

La relación entre todos los objetos de la escena queda

Un hijo puede ser, a su vez, padre de otros hijos, de forma que la jerarquía entre los objetos de una escena puede ser tan larga y compleja como nosotros queramos y necesitemos



Figura 5. La cinemática inversa es un método de animación más amplio que la directa. Aquí son los hijos los que arrastran a los padres.

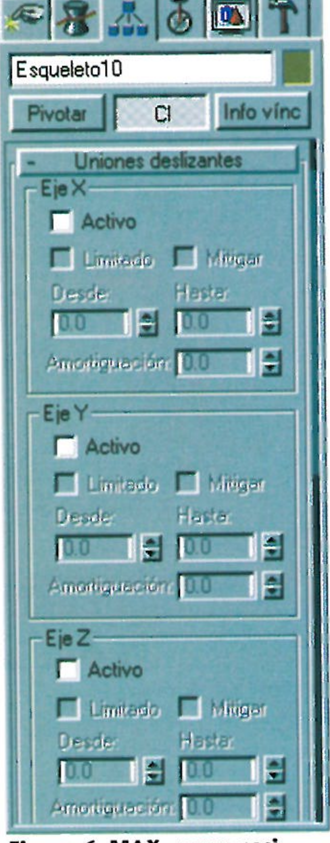


Figura 6. MAX, como casi todos los programas de modelado y animación, establece seis grados de libertad en los movimientos: tres deslizantes y tres rotacionales.

Para imponer las reglas de cinemática inversa a los objetos de una escena en 3DS MAX sólo tenemos que cubrir dos requisitos básicos: establecer con anterioridad todas las relaciones entre padres e hijos que reinan en el modelo, y activar el modo de cinemática inversa

perfectamente clara en la siguiente pantalla. Si ahora imprimiésemos un cierto giro a la base central, todo el conjunto giraría en el mismo sentido y la misma longitud de arco, conservando sus posiciones respectivas iniciales. Si en lugar de mover la base, giramos la plataforma, el resultado aparente es el mismo, PERO el padre (la base) sigue en su sitio.

La vinculación entre los objetos de una escena permite efectuar animaciones directas de grupos muy complejos con muy pocas instrucciones. Pensemos en un coche completo, con miles de objetos relacionados entre sí (tornillos, cables, ruedas, engranajes, etcétera). Si estableciésemos adecuadamente un árbol de relaciones jerárquicas padre/hijo entre todos estos objetos, estableciendo como primer padre el chásis del coche, bastaría con mover éste elemento para que las miles de piezas que componen el coche se desplazaran en su totalidad, conservando sus posiciones relativas.

Al mismo tiempo, como el movimiento de los hijos no afecta a los padres, es posible especificar el giro de una rueda o del volante en un momento determinado, independientemente del movimiento general del vehículo.

Cinemática inversa

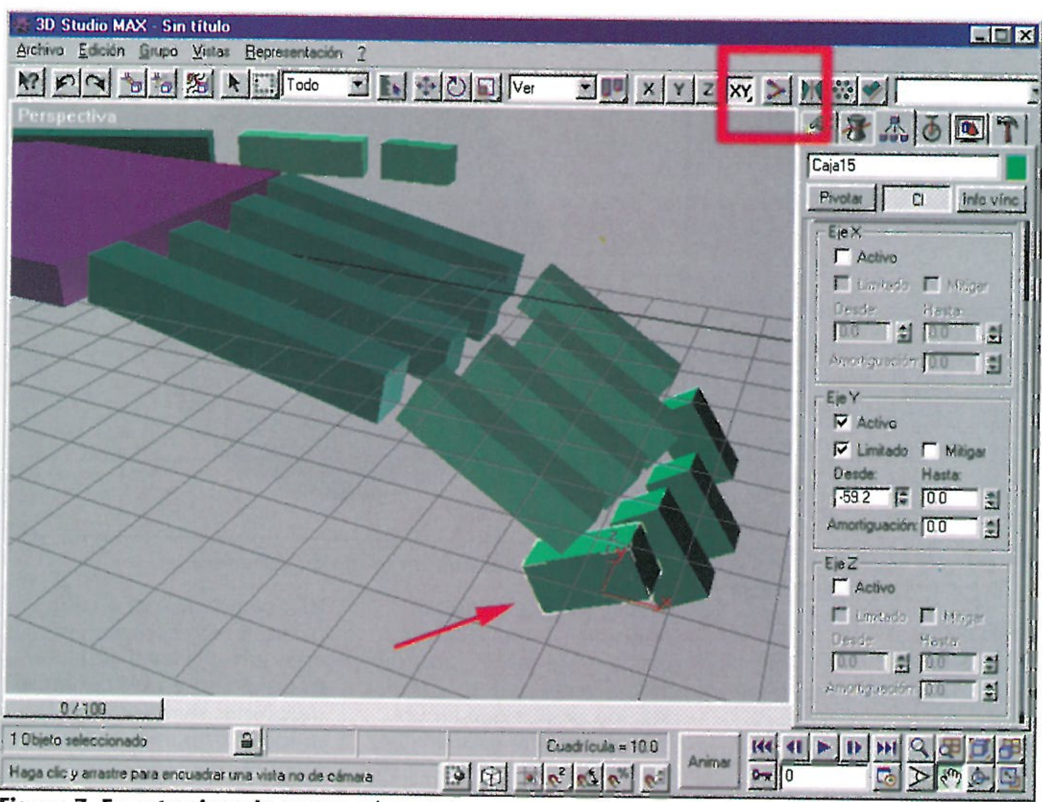
Aunque potente, el método de animación por cinemática directa tiene sus limitaciones y muchas veces ocurre (tanto en nuestra escena 3D como en la realidad) que son los hijos los que le dicen al padre dónde tiene que ir. Sin recurrir a ejemplos retorcidos (como cuando el niño tira algo al suelo y los padres salen corriendo a ver qué ha sido), fijémonos en nuestra mano derecha, y hagamos un experimento.

Apoyemos la mano sobre la mesa que tengamos delante o sobre las rodillas, para a continuación sujetar el dedo índice con la mano izquierda y tirar de él hacia arriba. Si no queremos sacarnos el hueso de su sitio, terminaremos por ceder al movimiento y la mano se levantará poco a poco *arrastrando al resto del brazo*. Hagámoslo de nuevo, pero más despacio. En un primer instante sólo se mueve el dedo índice, pero esto dura muy poco, sólo hasta que llegamos al límite de flexión de las falanges. En ese momento empieza a elevarse la mano, pero no el antebrazo. De nuevo, esto sólo dura un breve espacio de tiempo, hasta que agotamos la flexibilidad de este elemento de la cadena.

A este tipo de movimientos impuestos por los hijos a los padres de un esqueleto se le llama *cinemática inversa*, y es el principio según el cual se producen una gran cantidad de movimientos a nuestro alrededor.

Para imponer las reglas de cinemática inversa a los objetos de una escena en 3DS MAX sólo tenemos que cubrir dos requisitos básicos: establecer con anterioridad todas las relaciones entre padres e hijos que reinan en el modelo, y activar el modo de cinemática inversa. Lo primero es lo que hemos venido estudiando hasta ahora. Lo segundo es muy fácil y basta con pulsar en un icono situado aproximadamente a la mitad de la barra de herramientas superior, identificado por tres esferas pequeñas unidas. Al pulsarlo se vuelve de color azul, lo que indica que el modo de cinemáticainversa está activado.

Si ahora movemos una de las esferas que habíamos colocado en los extremos del tiovivo anterior, veremos que todo el conjunto de esferas y cilindros se mueve *arrastrados* por el movimiento de la bola. Sin embargo, hay que anotar dos circunstancias importantes; primera, el resto de bolas colgantes sólo se desplazan en un sentido u otro acompañando solidariamente a la plataforma superior; y segunda, no tenemos ningún control sobre el



En el apartado de uniones deslizantes hemos de asegurarnos de que ninguno de los tres ejes está activo, pues las falanges nunca se deslizan entre sí, sino que se limitan a rotar sobre sus articulaciones.

Figura 7. En este ejemplo vemos cómo debemos especificar las limitaciones de movimiento de cada uno de los objetos de la jerarquía. MAX tiene la ventaja de que tiene una interfaz muy intuitiva para este trabajo.

movimiento de los objetos del tiovivo. Tanto es así, que al cabo de unos pocos instantes es posible que todos los objetos que lo componen estén repartidos sin sentido por la escena.

Al contrario de lo que nos pueda parecer, la cinemática inversa es más fácil de controlar que la directa, pero exige un mayor conocimiento de lo que estamos haciendo. Aquí es donde nos va a ser útil ese estudio de movimiento al que hacíamos referencia, ya que el siguiente paso en el proceso de animación consiste en imponer *restricciones* de movimiento a todos los objetos de la jerarquía. Es evidente que sólo podemos flexionar las muñecas hasta un cierto ángulo, a partir del cual nos fracturamos la articulación. ¿Cómo definiríamos este movimiento entre dos objetos?

Hay dos tipos de movimientos fundamentales: deslizantes y rotacionales. Podemos ver los paneles que sirven para controlar cada uno de estos tipos de movimiento en las ilustraciones de este artículo.

En la última imagen podemos ver un modelo muy sencillo de una mano en la que hemos definido todas las jerarquías de padres e hijos. Ahora hay que especificar, una por una, las restricciones de movimiento de todas las falanges, para lo que accedemos al menú de cinemática inversa y seleccionamos la falangeta del dedo meñique. En el apartado de uniones deslizantes hemos de asegurarnos de que ninguno de los tres ejes está activo, pues las falanges nunca se deslizan entre sí, sino que se limitan a rotar sobre sus articulaciones.

Por tanto, lo que sí deberemos hacer es activar la unión rotativa en el eje Y, que es el que especifica la rotación horizontal de este hueso de la mano. Si lo dejamos así, este objeto tiene una libertad de movimientos de 360 grados alrededor de su centro geométrico, por lo que debemos limitarlo. En cada una de las secciones correspondientes a los tres ejes de rotación encontramos casillas para especificar los valores numéricos *desde* y *hasta*. Es más fácil de lo que parece y se trata de un proceso casi intuitivo. Basta con pulsar sobre una de las flechas y desplazar el ratón arriba y abajo. El efecto sobre el objeto es inmediato y podemos definir los valores límite de movimiento de una forma completamente visual.

Claro está que para trabajos más serios es probable que tuviéramos que ser más precisos, pero por el momento es un buen ejercicio. Los lectores "avispados" se habrán dado cuenta de que el centro geométrico de cada falange no está en el extremo del objeto, sino en su centro, ya que no lo hemos modificado desde que lo creamos. Esta es una de las tareas que dejamos pendientes para el mes que viene. En Lightwave y Softimage es muy sencillo modificar la posición de este punto pero, ¿cómo se hace en MAX?

Rafael Morales 3D

Fe de erratas

En el título del anterior artículo aparece la palabra "sicilio" en lugar de "silicio". El titular correcto habría sido "Las nuevas estrellas de silicio". Desde aquí, pedimos disculpas a todos nuestros lectores por la errata.

El Render

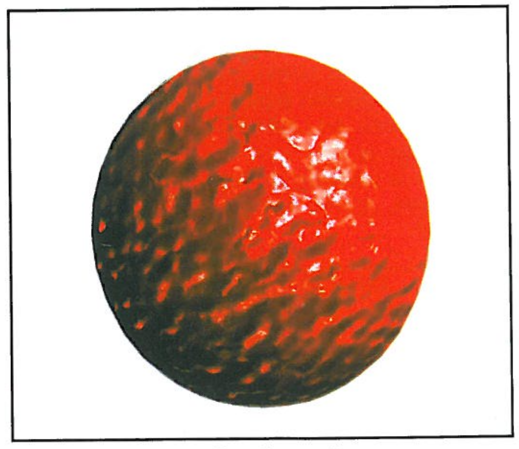
Este mes intentaremos explicar cuál es el proceso de render, con la idea de clarificar algunos conceptos que nos pueden llevar a error. Además, vamos a introducirnos no sólo en los problemas de software sino también en los de hardware y todas las posibles soluciones para acelerar este proceso: render-farms, ordenadores multiproceso, ordenadores con varios procesadores, etc...



Ejemplo de render con un mapa de reflexión.

El error más habitual es la mala interpretación de la frase "render en tiempo real" aplicada a cada una de las diferentes posibilidades, como pueden ser: videojuegos en tiempo real, programas que "renderizan" en tiempo real, visualización en tiempo real,...

El problema no reside sólo aquí sino en todos los conceptos asociados a la palabra render, y que nos sugieren preguntas del estilo: ¿Cuánto tarda en renderizar un PC?, ¿y una Silicon Graphics?, ¿y una estación gráfica profesional?, ¿por qué tarda a veces más en una estación gráfica que en un PC?, ¿y por qué pasa a veces al revés?, ¿qué ordenador es mejor para renderizar?, ¿qué dispositivos de hardware necesitamos?, ¿qué es lo que influye para que un render tarde más o menos?, ...y un sinfín de preguntas que se pueden resolver si se tuvieran, desde el comienzo, claros una serie de conceptos teóricos que implican el conocimiento de los factores que influyen en el proceso de render que son tanto de software como de hardware.



Una esfera tras aplicarle un *Bump*.

Para empezar a aclararnos vamos a dar una definición de la palabra render: un render es un proceso mediante el cual el ordenador calcula una imagen, basándose en los datos que obtiene de la escena 3D - luces, materiales, texturas - que son aplicados a los objetos presentes en ella.

¿Cómo funciona?

Una vez que hemos definido los componentes de la escena 3D, la cantidad de luces con sus características de color, posición y la opción de crear sombras, materiales y texturas aplicadas a nuestros objetos (superficies), el ordenador, al darle la orden de render, divide las superficies en triángulos, proceso conocido como *tessellation*, para, posteriormente, calcular cómo le afecta a cada triángulo los componentes anteriores y devolver la información en un conjunto de píxeles que, todos ellos unidos, formarán nuestra imagen.

Por su parte la definición de tiempo real es cuando el ordenador es capaz de calcular la imagen a la velocidad que necesitemos para nuestra visualización. Si nosotros necesitamos ver una animación en tiempo real, recordemos que un segundo de vídeo son 25 imágenes, nuestro ordenador debería ser capaz de calcular 25 imágenes en un segundo para, posteriormente, visualizarlas. En cambio, tiempo diferido es cuando el ordenador toma más tiempo en calcular las imágenes que el que necesitamos para visualizarlas. Si una imagen es compleja puede llegar a tardar hasta 2 o más horas en ser calculada.

Factores que influyen en el tiempo de Render

Hay que distinguir entre los factores que son propios de la definición de la escena y los que dependen de los parámetros que le demos al render. Así, según esta clasificación podemos tener factores propios de la escena que, a su vez, están divididos en los siguientes:

1.- Luces: Afectan al modo en que se ven los objetos en

la escena 3D y pueden cambiar totalmente la apariencia de éstos con modificar alguno de los parámetros. También están divididos en:

- Cantidad: a mayor número de éstas, más lento será el proceso
- Tipo: cuantas más características o parámetros tenga, más complejo será el cálculo.

Asimismo, las luces pueden ser Infinitas (*Infinite*), dirección y color, todos los objetos reciben la luz con el mismo ángulo. Por ejemplo, el Sol; Puntuales (*Point*), posición, color y zona que ilumina; Luz omnidireccional, los rayos salen desde ésta en todas las direcciones y según sea la posición de los objetos, estos recibirán la luz con diferente ángulo. Por ejemplo, una bombilla; Focos (*Spot*), posición, color, zona que ilumina, ángulo de apertura y ángulo de penumbra; y, finalmente, Luz unidireccional, que tiene una forma de cono con lo cual podemos definir zonas muy concretas de iluminación. Por ejemplo, una linterna.

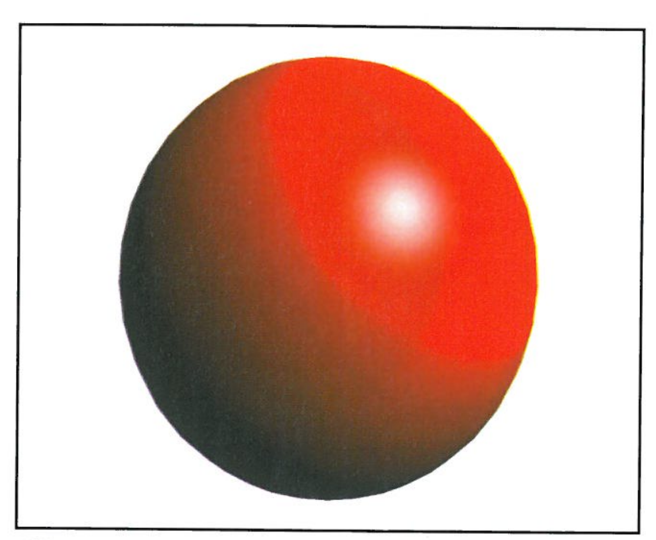
Además, las luces tienen otros parámetros que pueden ralentizar el tiempo de render notablemente, como son: Volumétricas, sensación de volumen en las luces (linterna), Flares, simulación de destellos en la cámara, y Halo y Glow, efectos especiales.

- Sombras(Asociadas a las luces): sirven para dos cosas, principalmente, para dar sensación de profundidad y para crear más realismo en la escena. Es una de las características que más ralentiza el render, sobre todo, si el resultado que queremos obtener es de una sombra difuminada.

Hay que evitar, en la medida de lo posible, que objetos que no necesiten sombras las tengan (objetos que no se vean, objetos demasiado pequeños, objetos sin importancia en la escena..). Dos grandes tipos de sombras que comparten casi todos los programas son las generadas por *Deph Map* y por *Raytracing*. En cuanto al primero, usan un algoritmo modificado de Z-buffer para crear sombras más rápidas pero no tan precisas como las de *Raytracing*. Este algoritmo calcula el color y la profundidad (z) de cada píxel basándose en su superficie y la distancia desde la cámara.

Antes de comenzar el render, un *Deph Map* es generado para la luz, y contiene la información sobre la escena desde la perspectiva de la luz; esta información describe la distancia desde la luz hasta los objetos de la escena y el color de la sombra sobre ellos. Durante el proceso de render, el mapa es usado para determinar si un objeto está en sombra o no. Por su parte, *Raytracing* conlleva el cálculo de los rayos de luz que son reflejados, refractados y obstruidos por las superficies. Aunque se obtienen resultados muy realistas, es un método costoso en tiempo. Con este tipo de sombras podemos definir la cantidad de ésta que se mezcla con el color del objeto sobre el que va a aparecer para obtener un resultado más realista.

II.- Geometría: Cuanta más información tengan los objetos de la escena más tardará la imagen en ser calculada, por ello hay que intentar simplificar al máximo la cantidad de polígonos y curvas de nuestros objetos. Un concepto importante es el de las normales de los objetos; cada triángulo de mi objeto es una superficie planar que tiene una cara frontal orientada en una dirección. La orientación de esta superficie es mostrada por un vector conocido como normal, localizado en los vértices de cada triángulo. Como cada polígono tiene un solo lado visible por la cámara, la normal siempre necesita estar orientada en la dirección de la cámara para hacer la superficie visible, también se puede hacer que todas las caras de un objeto sean visible, un objeto transparente, lo que



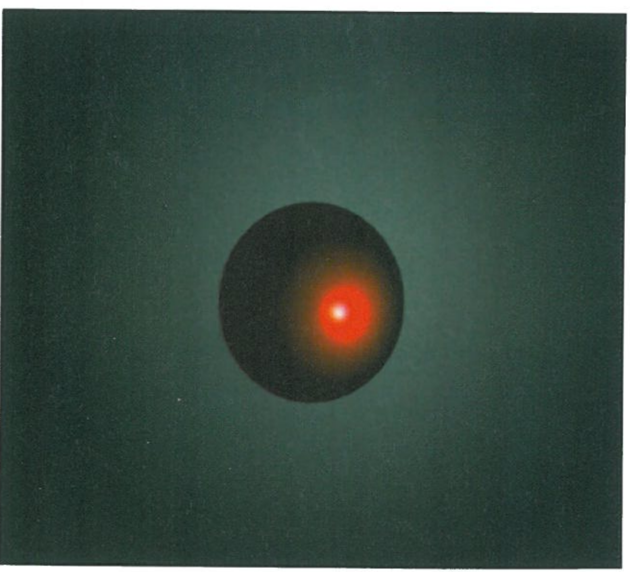
Objeto renderizado con sombreado *Phong*.

ralentizará notablemente el tiempo de render.

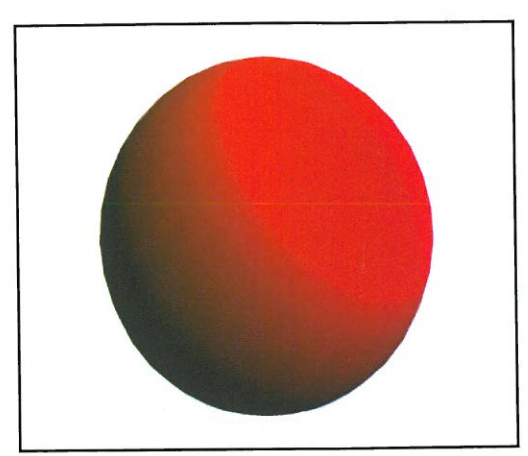
III.- Texturas: Las tiene en cuenta a la hora de calcular la apariencia de los objetos y lo que también en importante es que las tiene que cargar en memoria, con lo que si tenemos diez imágenes que usamos como texturas que ocupan 1 Mb cada una, el ordenador usará 10 Mb para almacenarlas en la memoria, viendo disminuida la cantidad de memoria disponible para realizar el render.

IV.- Materiales: Dan apariencia a nuestro objeto, basándonos en una serie de parámetros como transparencia, color, reflexión,... Los materiales se pueden dividir en varias clases:

Un *Deph Map* es generado para la luz, y contiene la información sobre la escena desde la perspectiva de la luz; esta información describe la distancia desde la luz hasta los objetos de la escena y el color de la sombra sobre ellos

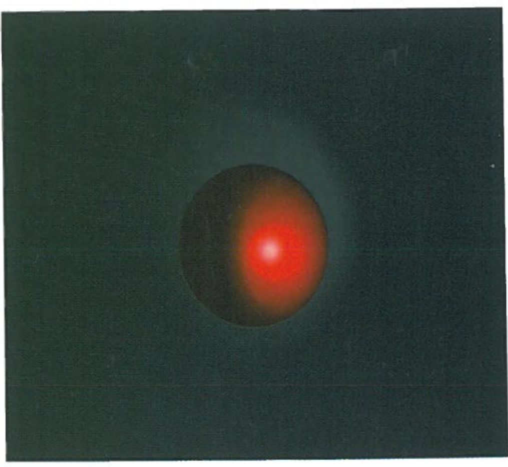


Render de una esfera con luz puntual.



Aquí podemos ver una esfera renderizada con luz ambiente.

- Tipos: cuanto más complejo sea el material más tardará en renderizar. Dentro de los tipos de materiales se encuentran estos otros: *Blinn*: usa cuatro zonas de iluminación, difusa, especular, índice de refracción y excentricidad. Este tipo resulta igual que el *Phong*, exceptuando la forma y brillo de la zona especular. *Phong*: utiliza las zonas de iluminación de ambiente, difusa y especular. Crea un sombreado suave y genera un punto especular al calcular la relación entre las normales, la luz y el punto de vista de la cámara. *Lambert*: tiene en cuenta la intensidad de los rayos reflejados. Interpola entre las normales para obtener un sombreado progresivo dando una apariencia mate a la superficie. No tiene especular pero si zonas difusas y de ambiente. *Constant*: ignora la orientación de las normales de la superficie dando la apariencia de no tener ningún tipo de sombreado, como un color plano.



Luz sport aplicada a un render.

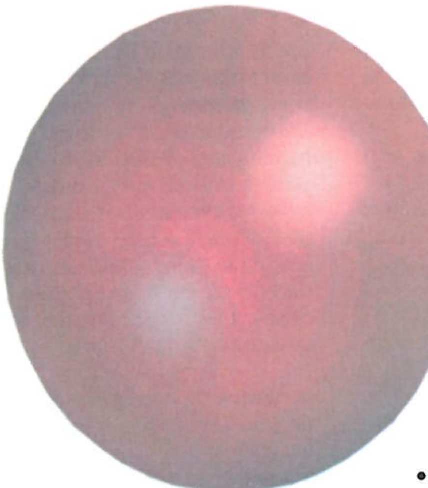
- Cualidades de los materiales: éstas, a su vez, se pueden clasificar en: Transparencia, da apariencia transparente a la superficie; Reflexión, calcula cómo rebotan los haces de luz sobre los objetos y cómo interactúan entre ellos para simular el reflejo; Refracción, cuando un objeto es transparente podemos dar un índice de refracción que servirá para indicarle al ordenador cuánto queremos que se desvíe la luz al entrar en contacto con la superficie (objeto deformado debajo del agua); y Bump, simula superficies rugosas perturbando las normales de ésta.

Según la complejidad del material más retrasaremos el proceso de render.

Parámetros del Render

En cuanto a los parámetros del render, existe otra clasificación que está dividida en: *Antialiasing*: Sirve para evitar el efecto dentado de los objetos; utiliza un algoritmo matemático que subdivide la zona del píxel para suavizar este efecto. Cuanto más grande sea la imagen menor será el problema.

- Campos (*Fields*): Cuando en nuestra escena existen movimientos muy rápidos de objetos o de la cámara, nos vemos obligados a renderizar por campos para obtener dos imágenes separadas de cada uno de los campos, el par y el impar, que, posteriormente, entrelazaremos antes de pasar a vídeo. Esto evita el efecto estrobo.



- *Motion Blur*: Efecto que simula la estela que deja un objeto al pasar delante de la cámara a alta velocidad.

- Tamaño de imagen: Cuanto más grande más lento.

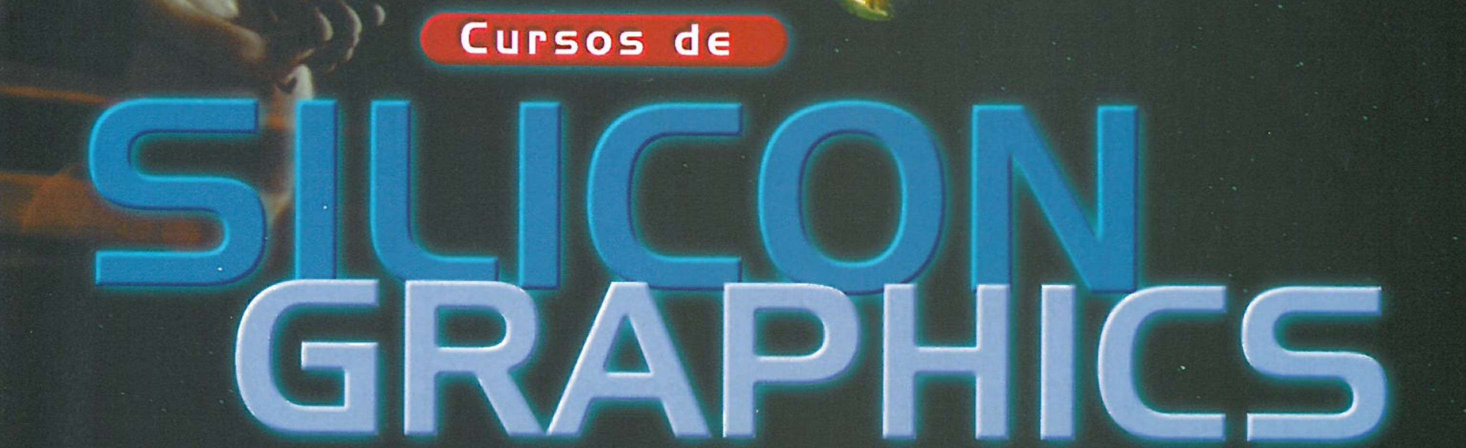
Equipamiento

Dentro de este apartado definiremos tres grupos claramente separados: los ordenadores caseros, comúnmente conocidos como PC´s, las estaciones de trabajo y los supercomputadores, dando una referencia aproximada del precio para hacer posteriormente un cálculo que nos facilite la relación velocidad/precio relativa a cada equipo.

Intentaremos explicar cada uno de estos grupos dirigiéndonos hacia la aplicación y el rendimiento que se les saca a la hora de realizar un render, separando, claramente, cada uno de ellos según sean nuestras necesidades de tiempo.

NOTA: La diferencia entre el presupuesto más económico y el más caro dependería de la marca del monitor, de la tarjeta gráfica y de los componentes como disco duro, memoria, etc.

Según nuestras necesidades de velocidad a la hora de renderizar deberemos optar por un equipamiento cada vez más alto, aunque es difícil obtener una regla con la que obtengamos una media que nos permita decidir qué máquina comprar y cuánto tardará en renderizar una imagen.


Cursos de
SILICON
GRAPHICS

I.- PC´s: (Gama media):

- Pentium II 166 Mhz
- 32 Mb de RAM
- 2 Gb de disco duro
- Tarjeta gráfica 2 Mb
- Monitor de 17´ pulgadas

Precio estimado:
200.000 ptas.

II.- Estaciones de trabajo:

A) BASADAS EN PENTIUM:

PC:(Gama media-alta)
- Pentium II Doble 200 Mhz
- 96 Mb de RAM
- 4 Gb de disco duro
- Tarjeta gráfica 4 Mb con aceleración Open-GL
- Monitor de 21´ pulgadas

Precio estimado:
400.000-750.000 ptas.

PC:(Gama Alta)
- Pentium II Doble 300 Mhz
- 128 Mb de RAM
- 9 Gb de disco duro
- Tarjeta gráfica 8 Mb con aceleración Open-GL
- Monitor de 21´ pulgadas

Precio estimado:
750.000 a 1.500.000 ptas.

INTERGRAPH:(Gama media-alta)
- Pentium II Doble 200 Mhz
- 96 Mb de RAM
- 4 Gb de disco duro
- Tarjeta gráfica 4 Mb con aceleración Open-GL
- Monitor de 21´ pulgadas

Precio estimado:
750.000 a 1.500.000 ptas.

INTERGRAPH:(Gama Alta)
- Pentium II Doble 300 Mhz
- 128 Mb de RAM
- 9 Gb de disco duro
- Tarjeta gráfica 8 Mb con aceleración Open-GL
- Monitor de 21´ pulgadas

Precio estimado:
1.500.000 a 3.000.000 ptas.

B) BASADAS EN ALPHA:

- ALPHA 300 Mhz
- 96 Mb de RAM
- 4 Gb de disco duro
- Tarjeta gráfica 4 Mb con aceleración Open-GL
- Monitor de 21´ pulgadas

Precio estimado:
750.000 a 2.000.000 ptas.

C) BASADAS EN MIPS: (SILICON GRAPHICS)

O2: (Gama media-alta)
- R5000 a 200 Mhz
- 128 Mb de RAM
- 4 Gb de disco duro
- Tarjeta gráfica 2 Mb
- Monitor de 21´ pulgadas

Precio estimado:
1.500.000 ptas.

Octane: (Gama media-alta)
- R10000 a 200 Mhz
- 128 Mb de RAM
- 4 Gb de disco duro
- Tarjeta gráfica 2 Mb
- Monitor de 21´ pulgadas

Precio estimado:
3.500.000 a 7.000.000 ptas.

III.- Supercomputadores:

Debido a la poca familiarización del lector con estos ordenadores únicamente mencionaremos las diferentes empresas con sus modelos respectivos (si procede).

a) Silicon Graphics y Cray:
- Onyx, Challenger.

Precio estimado:
15.000.000 a
100.000.000 ptas.

b) HP, IBM, SUN
Precio estimado:
No disponible.

MENCIÓN ESPECIAL

RENDER-FARMS:
a) En un rack:
Es una instalación de hardware en la que están presentes un número determinado de procesadores (normalmente entre cuatro y ocho Pentium II 200 Mhz), en la que estos procesadores pueden compartir una serie de recursos de hardware: tarjeta gráfica, memoria, disco duro y monitor, para aprovechar al máximo la potencia de proceso.
b) Unión en red:
Se tendrían un número determinado de ordenadores (con su configuración de hardware individual) que, mediante una conexión en red y el software apropiado, se pueden repartir las tareas de render.
Precio estimado: 3.500.000 a 7.000.000 ptas.

Javier Viñambres 3D



Soluciones profesionales

POWERLOOK 3000

La calidad del tambor en plano.
Basado en un diseño de doble lente, la primera lente del PowerLook 3000 es capaz de escanear opacos y transparencias hasta 216x297 mm en modo normal a 1220x3048 dpi. La segunda puede aumentar pequeños formatos con una resolución de 3048x3048 dpi, (86x297 mm). Si le sumamos un rango de densidad de 3,6D y la tecnología BET de 42 bit de profundidad de color, tenemos lo que todo profesional desea: Calidad a precio competitivo.

- Una pasada, CCD color.
- Diseño de doble lente.
- Resolución óptica: Lente 1, 1220x3048 dpi
  Lente 2, 3048x3048 dpi
- Resolución máxima: 12192x12192 dpi
- 42 bit, billones de colores.
- Rango de densidad: 3,6D.
- Adaptador de transparencias incorporado.
- binuscan PhotoPerfect Master incluído.







MIRAGE II/IIse

Escáner profesional tamaño A3.
Cuando el tamaño es una necesidad, esta serie de escáneres nos permiten tener un formato de trabajo de A3 sin renunciar a la resolución necesaria para escanear diapositivas, **Mirage II**. Si lo prioritario es disponer de este gran formato la serie **Mirage IIse** nos ofrece una relación calidad/precio inmejorable.

- Una pasada, CCD color.
- Diseño de doble lente en versión Mirage II.
- Resolución óptica: Lente 1, 700x1400 dpi, Mirage II, Mirage IIse.
  Lente 2, 1400x2800 dpi, Mirage II.
- Resolución máxima: 9800x9800 dpi.
- 36 bit, billones de colores.
- Rango de densidad: superior a 3,3D.
- Adaptador de transparencias incorporado.
- binuscan PhotoPerfect Master (Mirage II), Advanced (Mirage IIse).





POWERLOOK III

Escáner profesional de Alta Resolución.
Con este escáner no tendrá que enviar cada fotografía o imágen a un servicio de tambor, esta no es la solución para tener imágenes de calidad. Con el PowerLook III tendrá una calidad profesional en todas sus reproducciones, tanto en opacos como en transparencias.

- Una pasada, CCD color.
- Resolución óptica: 1200x2400 dpi.
- Resolución máxima: 9600x9600 dpi.
- 42 bit, billones de colores.
- Rango de densidad: superior a 3,4D.
- Área de escaneado en opacos: 210x297 mm, transparencias: 216x254 mm.
- Adaptador de transparencias incluído.
- binuscan PhotoPerfect Master.









¡Es perfecto! ¡Es simple!... ¡El mejor valor añadido!
binuscan con la tecnología ReCo interpreta los datos en crudo que entrega el escáner, los analiza con su proceso de inteligencia artificial y crea nuevos pixels que ningún otro filtro o proceso puede detectar para tener una nueva imágen perfecta.



DISVENT, S.A. Avda Josep Tarradellas, 46 tda. 08029 BARCELONA Tel. (93) 321 50 14 Fax (93) 322 68 06 email: dvcom@disvent.com

Smacker, la gran alternativa

Si bien el formato AVI ha sido el líder indiscutible en vídeo para PC, la llegada de nuevas alternativas como el MPEG u otros tipos de compresión hacen que el formato AVI no sea quizá la plataforma ideal para expresar nuestras ideas electrónicas.

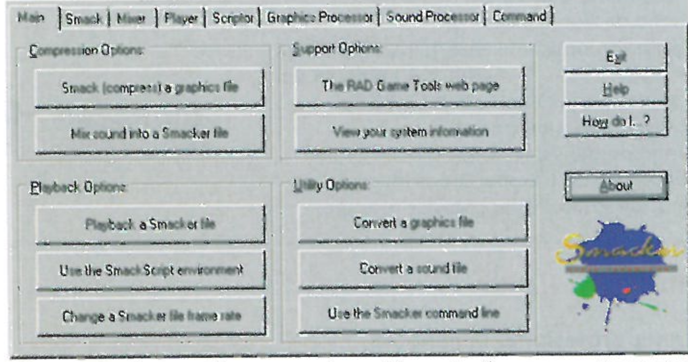


Figura 2. Convertir una animación o imágenes a formato SMACKER.

Desconocida para la mayoría de la gente, la compañía *RAD Game Tools* se dedica a servir profesionalmente sus productos a las casas desarrolladoras de videojuegos y obras Multimedia. Afincados en Salt Lake City (Utah, EE.UU), el equipo lo componen seis personas dirigidas por Jeff Roberts, Ken Coburn y Mitch Soule.

Sus dos productos estrella son Smacker, la seria alternativa al formato AVI con un buen montón de poderosas razones para serlo, y el MILES Sound System, un sistema de sonido de calidad soberbia. Ni que decir tiene que cerca de 600 compañías utilizan estos dos sistemas y parecen estar encantados. Pero... ¿por qué?

Para añadir nuestras músicas/locuciones, tan sólo tenemos que especificar el fichero SMK, decirle en qué pista irá el audio, una serie de caracteristicas tecnicas obvias. Tras esto, podremos añadirlo sin problemas. Sin duda alguna, el uso de siete pistas de audio hace que podamos mezclarlas en tiempo real de manera inteligente

Características generales

Al ver por primera vez el programa, la impresión que nos da es un tanto fría, ya que tenemos delante una aplicación donde no prima mucho lo visual. Muchos apartados técnicos, muchos botones y muchas pestañas que tocar. Pero no hay que preocuparse de ello, ya que facilmente podremos realizar nuestro primer intento (fallido o no, pero intento al fin y al cabo) de crear un vídeo. Pero lejos de nuestra primera impresión, Smacker nos ofrece un trato amigable y profesional al mismo tiempo. Si dejamos el cursor quieto por algunos de los botones, nos dará ayuda suplementaria, y podremos acceder a otras opciones.

El menú principal da paso a las diferentes funciones del programa, que serán aquí explicadas:

En la figura 2, a la izquierda del todo, tenemos dos grandes ventanas donde elegiremos nuestra fuente de vídeo, que puede ser AVI, FLIC, SMK (el formato propio de Smacker), DV, MOV, JPG, TGA, TIF, PNG, PCG, SGI, archivos de sonido y especiales. Si pulsamos con el botón derecho podremos copiar, pegar, borrar, renombrar y crear listas de ficheros, ahí es nada. Y además, si pulsamos dos veces en un archivo, podremos verlo. Esto se aplica a TODAS las ventanas donde veamos que podemos elegir ficheros para importar.

En el apartado *Options* es donde más trabajo realizaremos. En las cinco pestañas que tenemos podremos establecer los ratios de compresión del vídeo final, el tipo de compresión (por campos o no), la paleta, el control de frames, el tipo de sonido y su calidad, y un modo experto donde podremos elegir una serie de frames y decir a cuántos Kbps se comprimirán (genial para partes de vídeo que requieran más KBps por segundo por su calidad.

Una vez rellenadas las casillas correspondientes (pocas tendremos que

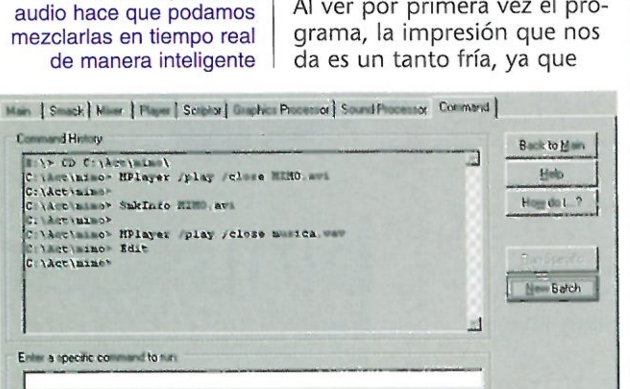


Figura 1. Aquí tenemos el menú donde se organiza todo.

Soporte y más soporte

Muchas compañías utilizan el sistema de vídeo de Smacker. Entre ellas se podrían citar Accolade, Activision, Animatek, Atari Games/Midway, Auran, BMG, Blue Byte, Bullfrog, Codemasters, Core Design, Digital Domain, Electronic Arts, Epic, GT Interactive, Infogrames, Krisalis, Microprose, Ocean... La lista es realmente interminable.

al formato AVI

tocar, porque por defecto está todo listo para comprimir), pulsaremos el botón *Smack* que viene en la zona derecha, y nos iremos de fiesta o a dormir, porque el proceso es tremendamente lento con vídeos grandes .

Echando un vistazo a la figura 3 ya tenemos nuestro vídeo, pero... ¿qué hay del sonido? Pues en este apartado tendremos la posibilidad de añadir hasta siete bandas sonoras a nuestro vídeo, con lo cual tendremos un vídeo multipista genial. El propósito inicial puede ser para, por ejemplo, tener diferentes idiomas dentro del vídeo.

Y para añadir nuestras músicas/locuciones, tan sólo tenemos que especificar el fichero SMK, decirle en qué pista irá el audio, una serie de caracteristicas tecnicas obvias. Tras esto, podremos añadirlo sin problemas. Sin duda alguna, el uso de siete pistas de audio hace que podamos mezclarlas en tiempo real de manera inteligente.

Si deseamos ver un vídeo, no hay más que meterse en el apartado que aparece en

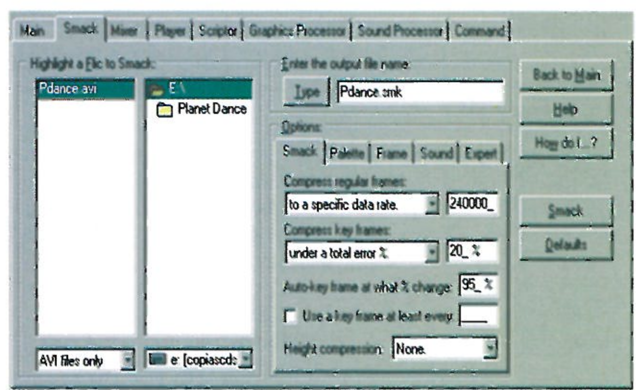


Figura 3. Insertar las bandas sonoras de nuestro vídeo.

Una prueba práctica

Vamos a hacer una prueba práctica para demostrar que todo esto que hemos explicado funciona y, además, muy bien. Convertiremos los dos tipos de ficheros, AVI y SMK, para ver las diferencias y las ventajas de cada uno de ellos.

Bien, para empezar elegiremos cualquier escena creada con cualquier programa de animación. Da igual la duración, aunque debería tener movimientos de cámara y efectos especiales, así como muchos colores para ver cómo afecta la degradación final, así que depende de cada uno de nosotros. Lo ideal son 100 frames o así, a 30 cuadros por segundo. Además, la generaremos en formato BMP o TGA, con una resolución de 320x240 o 320x200.

¿Ya está nuestra animación generada? Bueno, vamos primero a generar el vídeo en formato AVI. Los que siguieron el anterior curso de Premiere sabrán fácilmente cómo crear un AVI, aunque de todas formas no hay que utilizar Premiere obligatoriamente. Smacker permite convertir a AVI las imágenes. Sin embargo, es preferible utilizar Premiere, ya que Smacker tarda bastante más en generar el AVI, y es más impreciso que el primero.

Viendo las propiedades del fichero AVI obtenemos una información que nos ayudará a establecer la del fichero SMK: los Kb por segundo. En el caso de este AVI, han sido 590 Kbps. No se le ha forzado a unos Kbs determinados para ver cuánto da de sí cada formato. Haciendo pruebas, con 590 Kb por segundo la calidad era muy superior que a 170 Kb en el mismo tipo de AVI.

Bueno, ahora le toca el turno a Smacker. Nos iremos a la opción *Smack*, donde haremos una lista de ficheros a comprimir. Para ello, seleccionaremos todos los ficheros, pulsaremos con el botón derecho en la zona de ficheros, y crearemos una lista (viene la opción). Una vez creada, la cogeremos.

Ahora viene quizá la parte más importante, donde estableceremos los valores de compresión.

En el CD de portada, dentro del directorio \ARTIC\SMACK, podemos observar ambas animaciones. La animación es corta, dura tres segundos y poco más a 30 cuadros por segundo. Se han utilizado efectos de brillo para ver cómo afecta a la compresión final.

Tenemos tres campos con los que deberemos andar con cuidado para ver el resultado lo más parecido en ambos formatos. Los Kb por segundo los estableceremos en 590, en lugar de 240 como viene por defecto. El resto de apartados no los tocaremos.

Para establecer los cuadros por segundo iremos al apartado *Frame*, donde en el campo *Frame Rate Override* le pondremos 30. No activaremos el sonido, ya que no lo llevará. Sobre la paleta no tocaremos ninguna opción, pues las que vienen por defecto son las perfectas aunque, por si acaso, hay que verificar que los campos siguientes sean:

Use/Create Palette From: The Input File
Win System Colors: Never
Halftone Input Files: When HiColor
8-Bits Input Palettes: Create New

Con estos datos ya podemos pulsar el botón *SMACK!*, y nos dedicaremos a otra cosa, ya que el programa puede estar compilando mientras nos dedicamos a escribir o hacer otras cosas que no interfieran mucho en el rendimiendo del procesador.

¿Ya ha sido generado nuestro vídeo? Vamos a comparar la calidad de uno y otro, a ver qué obtenemos. En el cuadro referente a las Diferencias Técnicas podemos ver la información referente a un vídeo de alta calidad y otro de baja calidad. Igualmente, tenemos en el CD de portada cuatro vídeos, dos en formato AVI con calidades alta y baja e igualmente en formato SMK.

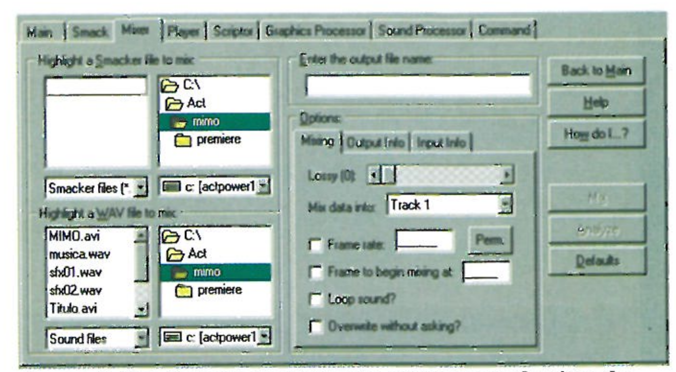


Figura 4. Ver ficheros en formato SMK con todo tipo de lujos y opciones.

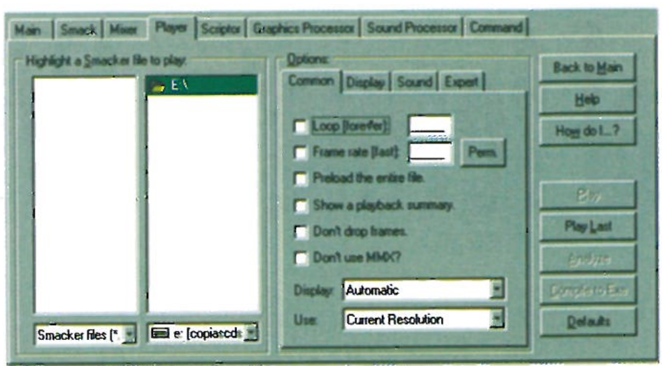


Figura 5. El exclusivo sistema de Scripts de Smacker nos permite hacer auténticas "virguerías" con los vídeos.

la figura 4. Aquí podremos elegir resoluciones, *loops*, bandas de audio a escuchar, ver un sumario del fichero, simular una cantidad de Kbps por segundo (ideal para CD-ROM), y una serie de apartados técnicos que nos serán muy útiles. Ni que decir tiene que el reproductor que viene por separado nos permite todo esto a través de comandos.

Pero no todo se reduce a observar y deleitarse con los vídeos que hemos creado, sino que podremos modificar su reproducción a voluntad. Mediante un sistema de programación de *Scripts* muy sencillo, podremos manejar a voluntad los ficheros SMK sin ningún problema. Aunque con el programa en sí no viene nada referente a los *Scripts*, en la ayuda viene algún que otro ejemplo.

Al igual que el audio, el vídeo también puede ser mezclado y editado una vez comprimido. Con el mismo apartado de *Mixer* podremos hacer todo esto, cambiar el *Frame Rate* de los vídeos y algunas otras características.

A veces es necesario convertir un montón de ficheros de formato. Incluso puede que tengamos varias aplicaciones con cada tipo de función. Bueno, pues ahora ya no, porque *Smacker* lo incluye de serie y, además, de paso, podremos crear la paleta que conformará el vídeo final, con lo que nos ahorraremos un paso más a la hora de comprimir.

Aquí haremos los mismo que hicimos con las imágenes, sólo que aplicado al sonido, cambiando los KHz, resolución y tipo. Ideal para sacar bandas de audio de animaciones.

La pregunta que nos hacemos es: ¿Por qué han añadido ésta función cuando Windows tiene su propia ventana de comandos?

Al igual que el audio, el vídeo también puede ser mezclado y editado una vez comprimido. Con el mismo apartado de Mixer podremos hacer todo esto, cambiar el Frame Rate de los vídeos y algunas otras características

Diferencias técnicas entre ambos formatos

Una vez procesados los archivos, éstas son las diferencias entre uno y otro formato en los dos tipos de calidades:

Calidad alta

El formato AVI tiene un flujo de 590 Kb por segundo, con lo cual la calidad de la imagen es muy buena y no se aprecian degradaciones. Está comprimido a 24 Bits de color, por lo que no se notan saltos importantes de color. Premiere comprimió este fichero en cerca de 2 minutos, y el espacio que ocupa en disco duro es de 1.94 MB. Sobre la calidad de reproducción, puede que alguna que otra vez se quede atascado, con la consiguiente frustación. Con la emulación 8 Bits a pantalla completa se ve una degradación bastante fuerte a nivel de color, y se observan bastantes puntos que hacen que ver el vídeo a pantalla completa sea frustante.

El formato SMK tiene un flujo de 500 Kb por segundo, pero la calidad general es mejor que en formato AVI. Los colores son más brillantes, y se oberva una cierta mejora en las partes más conflitivas. Está comprimido a 8 Bits, pero se ve igual que a 24. Smacker tardó 9 minutos en compilar el vídeo, lo cual significa una importante pérdida de tiempo, aunque bien justificada por la calidad. El espacio que ocupa en disco son 1.77 MB, un poco menos que su competencia AVI. La reproducción es perfecta, sin saltos ni nada por el estilo y a pantalla completa se ve igual que en ventana, y no se observan degradaciones por ningún lado.

Calidad baja

El formato AVI tiene un flujo de 170 Kb por segundo, así que la degradación es importante. Se ven manchas borrosas y la definición pierde bastante. Los mismos puntos generales son aplicables a este fichero, aunque ocupa en disco 643 Kb, menos del doble que el anterior.

El formato SMK tiene un flujo parecido, y la degradación también se hace notar, aunque la definición es bastante mejor y se notan detalles que en el anterio formato no se notaban. Ocupa en disco un poco menos que el formato AVI.

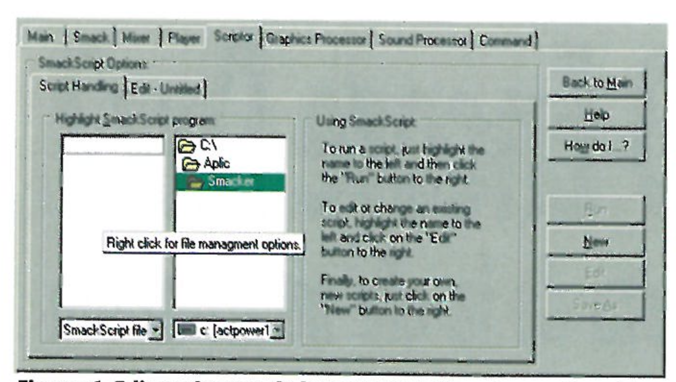


Figura 6. Editar algunas de las caracteristicas de nuestro vídeo.

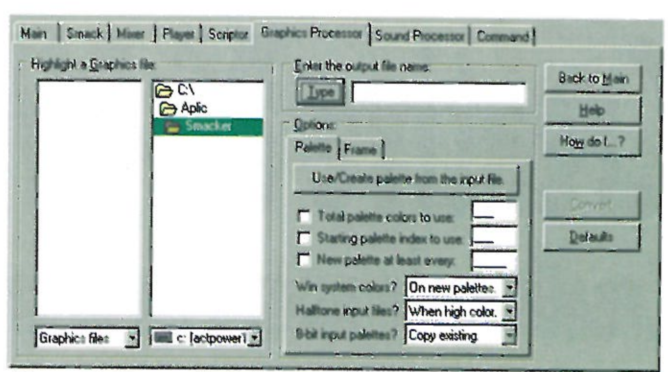


Figura 7. Convertir ficheros gráficos y de vídeo.

Bueno, la duda se nos despeja rápidamente cuando queramos comprimir por la noche un buen montón de vídeos uno detrás de otro, ya que gracias a esta ventana de comandos podremos copiar y pegar una serie de informaciones que sólo saldrán ahí. En la ayuda viene un apartado que explica cómo hacerlo paso a paso. Además, sirve como depuración a nuestros vídeos, ya que ofrece otra serie de informaciones útiles. Es una especie de "histórico" de todo lo que hemos realizado en la sesión.

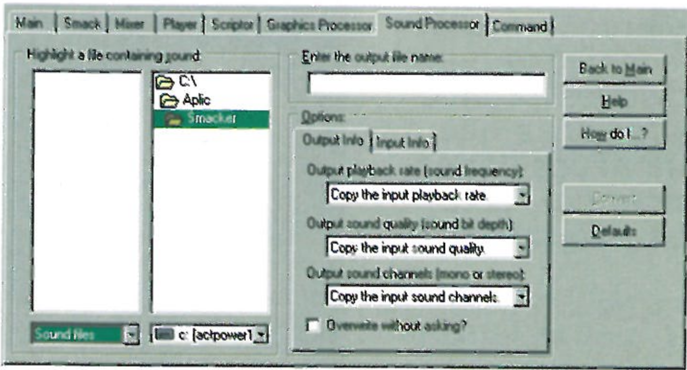


Figura 8. Convertir un fichero de sonido.

La decisión

Vistos los resultados de ambos formatos, la decisión es evidente. El formato SMK gana en muchos apartados, aunque quizá el más negativo sea que la compresión de ficheros con paleta de 24 Bits es bastante lenta, aunque su calidad sea superior incluso que los propios 24 Bits del formato AVI. No hay corrimientos extraños de color ni "basura" en forma de píxeles. La calidad de reproducción es envidiable en todo tipo de resoluciones y gana en aspectos técnicos como puede ser el tener siete bandas de audio, el poder establecer diferentes calidades dentro de un mismo vídeo, los útiles *Scripts* con los que podemos hacer maravillas con los vídeos y un sinfín de detalles que hacen que el formato SMK sea el ideal para desarrolladores de software.

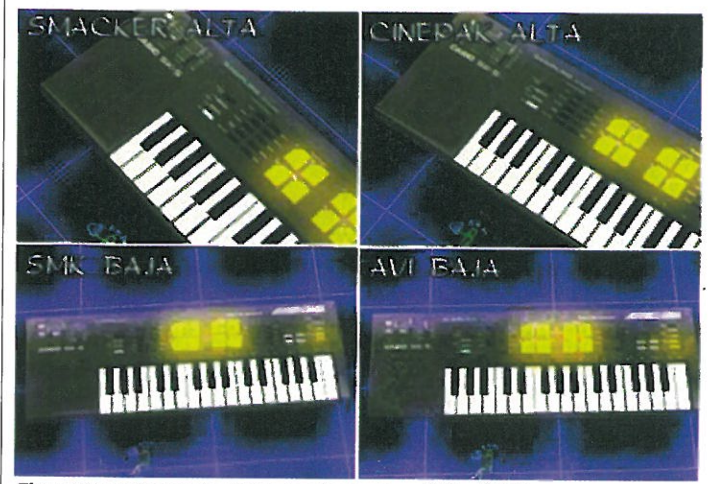


Figura 9. Diferencia entre los formatos AVI y SMK.

Vistos los resultados de ambos formatos, la decisión es evidente. El formato SMK gana en muchos apartados, aunque quizá el más negativo sea que la compresión de ficheros con paleta de 24 Bits es bastante lenta, aunque su calidad sea superior incluso que los propios 24 Bits del formato AVI

Todo esto, unido a su carácter multiplataforma y su versatilidad, ofrece un producto que posiblemente (y dependiendo de las casas desarrolladoras, claro) desbanque al formato AVI, e incluso al MOV.

Para saber realmente si esto es verdad, se han incluido en el CD de portada dos vídeos en formato AVI y SMK de diferentes calidades, así como el programa Smacker 3.1 completo, sin restricción. Si tenéis alguna duda, podeis consultarla en la dirección de E-mail: actpower@mx2.redestb.es

Antonio Casado 3D

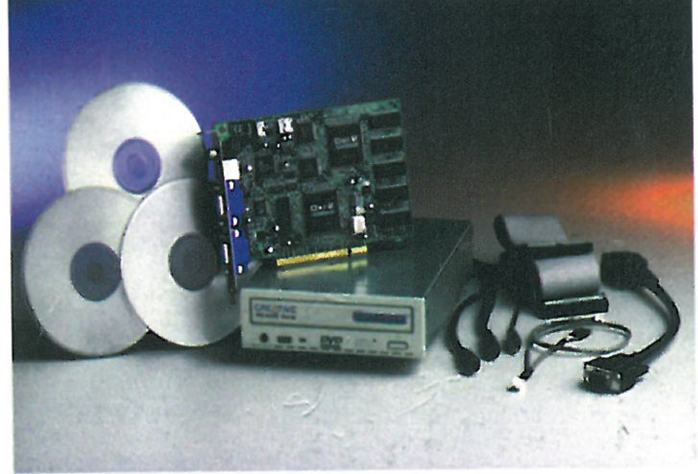
El sistema de ayuda

Si algo caracteriza a este genial programa es su sistema de ayuda. No sólo nos explicará a fondo cómo es el sistema SMK y nos dará un tutorial de cómo compilar como es debido nuestros vídeos en una de ellas, sino que en otro tipo de ayuda podremos pedirle una serie de preguntas (ya formuladas), para que nos aconseje sabiamente. En este caso, la ayuda es genial, fácil de comprender y acertada.

Creative PC-DVD Encore Dxr2

Creative, de nuevo en la cresta de la ola multimedia

La presencia de Creative Labs en el mercado multimedia y su capacidad para desarrollar nuevos productos con prestaciones que se adapten a las necesidades del mercado actual, está cada vez más extendida.



Su último producto multimedia, el lector DVD Encore Dxr2, nos muestra y su última tecnología de reproducción de imágenes: Dxr2 de Dynamic Xtended Resolution, atractivo que nos ayudará a decidir cuando llegue el momento de adquirir uno de estos dispositivos multimedia.

El sistema Dxr2 de Creative que, en realidad es una tarjeta PCI, se ha desarrollado con la idea de filtrar y eliminar efectos como el parpadeo y barridos horizontales como los que se producen en los televisores causados por la señal entrelazada a 30 imágenes por segundo.

Además Dxr2 es capaz, mediante su sistema inteligente de interpolación de líneas, de doblar la resolución y la velocidad de visualización, llegando a alcanzar una velocidad de hasta 60 imágenes por segundo. Sin pasar por alto su capacidad para detectar los cambios de velocidad en el muestreo de imágenes, evitándose así el denominado efecto de línea doble, y su potencia de cálculo a la hora de reescalar imágenes, en donde aplica un filtro especial, suavizando así la imagen en la nueva resolución sin perder calidad.

Sus características técnicas se mueven en el nivel medio de prestaciones: 2.7 Mb/s (2x DVD-ROM) y 3 Mb/s (20x CD-ROM), reproductor de vídeo DVD 1.0, MPEG-2 y MPEG-1 a 30 cuadros por segundo (NTSC), pantalla completa con resolución máxima de 1280x1024, sin olvidar la opción de superposición de vídeo analógico y la posibilidad de reproducción simultánea en TV y monitor de PC.

Y en cuanto a la reproducción de audio es de destacar su capacidad para reproducción en Dolby Digital (AC-3), sistema con el que se podrá disfrutar de alta fidelidad acústica de este sistema a través de los dos canales de la tarjeta de sonido de nuestro PC, e incluso disfrutar del sistema empleado en las salas de cine con el sistema receptor AC-3 de seis altavoces. Aparte de los requisitos de sistema que se muestran en el cuadro 1, para disfrutar del sistema Dolby Digital AC-3 5.1 Channel audio necesitaremos de un amplificador o receptor con entrada Dolby Digital.

Y en lo referente a software, Creative ha lanzado este producto con un completo paquete de siete productos entre los que destacan Wing Commander IV o Claw, y un

Requisitos del sistema

- Sistema operativo Windows 95
- Procesador Pentium o compatible a 100 Mhz o superior
- 16 MB de memoria RAM
- Tarjeta gráfica compatible SVGA 2MB de RAM
- 1 Slot PCI
- Conector Enhanced IDE con soporte para Bus Master DMA
- 1 slot de disco de media altura disponible para el lector DVD
- Tarjeta de sonido Sound blaster o 100% compatible
- Altavoces amplificados o auriculares

Formatos de disco soportados

- CD-R
- CD-Extra
- CD-ROM/XA
- Photo CD
- CD-RW
- DVD-Vídeo
- CD-Audio
- CD-ROM
- Video CD
- CD-I
- DVD-ROM

Conexiones

- Jack RCA S/PDIF de salida AC-3 5.1
- Conector de salida de audio JST
- Conectores para:
  Salida S-Vídeo
  Salida de vídeo compuesto
  Salida VGA DB-15
- Entrada VGA tipo DB-15

interfaz de navegación, el PC-DVD Player, un panel de control donde se incluyen un Pad direccional, una pantalla de estado, control de volumen, menús para personalizar datos de audio, color, idioma y botones de reproducción.

En los cuadros 2 y 3 se pueden observar las restantes características de este dispositivo, que, sin duda alguna y como siempre, plantea una seria competencia en su sector por una razón muy sencilla: el equilibrio entre prestaciones, software incluido y precio (47.000 ptas. aproximadamente).

Enrique Urbaneja 3D

PC-DVD ENCORE DXR2

Fabricante: Creative Labs
Aptdo. de Correos 38
08960 Sant Just Desvern, Barcelona
Tel: (91) 375-33-35
Fax: (93) 499-08-11
BBS: (93) 499-08-98/07-82
Web:
http://www.cle.creaf.com/europe/sp.html
Precio Aproximado: 47.000 Ptas.

POWERLOOK 3000



De la mano de Disvent nos llega este nuevo modelo de escáner de UMAX que fue presentado el pasado mes durante la última edición de Graphispag. Se trata de un explorador basado en un diseño de doble lente, de las cuales la primera es capaz de escanear opacos y transparencias hasta 216x297 mm a una resolución de 1220x3048 dpi, y la segunda es la encargada de aumentar los pequeños formatos con una resolución de 3048x3048 dpi, 86x297 mm. Y si a esto se le suma un rango de densidad de 3,6D y la tecnología BET de 42 bit de profundidad de color, se obtiene la calidad del tambor en plano a precio competitivo, lo que todo profesional desea.

Además del citado diseño de doble lente, entre sus características técnicas destacan también la exploración en una pasada, CCD color en un área de 216x297 mm para la primera lente y 86x297 para la segunda o un sistema de bandeja móvil para originales.

Asimismo, cuenta con un sistema de iluminación optimizado, diseñado especialmente para que difícilmente haya una pérdida de luz y la tolerancia a grandes ampliaciones sea menor que en cualquier otro escáner convencional. Incluye también un algoritmo super realzado de calibración con el que es posible detectar toda la información tanto de las áreas brillantes como de las oscuras, operación que ayuda a incrementar el rango dinámico y asegura una total precisión en las imágenes escaneadas en color.

En el apartado del software, el escáner incluye el driver MagicScan, que ofrece todas las funciones y controles necesarios para el profesional, como destramado, ajuste de niveles y de mejora del color, rotación y giro de imágenes, lectura de histograma, etc, con soporte de idioma y para plataformas Macintosh, Windows 3.1, 95 y NT. También incluye el software de post escaneado, y corrección y separación de las imágenes escaneadas Binuscan PhotoPerfect Master, basado en tecnología ReCo y que interpreta los datos escaneados y los utiliza para generar una nueva imagen balanceada, con un histograma perfecto. Por último, el paquete incluye también Live Picture en versión completa y Live Picture XT, dos herramientas para completar la producción del control y la manipulación de las imágenes con velocidad y calidad fotográfica.



Fabricante: UMAX
Distribuidor: Disvent S.A
Tel: (93) 321-50-14
Fax: (93) 322-68-06
Precio: No facilitado

Hardware

Monitor Apple MultipleScan 720



El Apple Multiple Scan 720 es un monitor de página completa de 17" (16" de diagonal del área visible), diseñado para los usuarios exigentes que buscan las prestaciones de una pantalla grande a un precio competitivo. Su tubo de imagen plano, con máscara de sombras ranurada, proporciona textos claros e imágenes definidas, convirtiéndolo en una opción ideal para el tratamiento de texto, las hojas de cálculo y la navegación por Internet.

Este monitor funciona con todos los ordenadores Macintosh y Power Macintosh, desde el más asequible hasta el más potente, e incluye el software Control Strip, que aparece automáticamente en pantalla y se utiliza para realizar precisos y sencillos ajustes de los controles del monitor, como la resolución y la profundidad de color. Y el tratamiento antiestático y antideslumbrante de su pantalla proporciona unas excelentes prestaciones de visualización a largo plazo. Tanto si se está pensando en actualizar el monitor más pequeño como si se prepara para una configuración totalmente nueva en el equipamiento, el monitor Apple Multiple Scan 720 es una opción inteligente, tanto por su flexibilidad en una gran pantalla como por su asequible coste.

Entre sus características cuenta con una pantalla plana de alta calidad, con máscara de sombras ranurada, que proporciona imágenes definidas. Incorpora un gestor del monitor en pantalla para realizar ajustes precisos de los parámetros e incluye el perfil de visualización ColorSync de Apple para obtener una precisa armonización del color entre las imágenes de pantalla y de la impresora. Asimismo, utiliza electrónica *Multiple Scan*, que proporciona una mayor flexibilidad de visualización y ofrece una gama de resoluciones hasta 1.280 por 1.024, e incluye software para conmutación de modo, para cambiar con sencillez los ajustes de resolución y proporciona capacidad multiplataforma (eso sí, se requiere adaptador VGA)

El monitor funciona con ordenadores Power Macintosh, Macintosh Performa y Macintosh Quadra con microprocesadores 68040 o posteriores que ejecuten el sistema operativo System 7.5 o posterior. También funciona con ordenadores que ejecuten Ms-DOS y Windows y utilicen VGA, pero eso sí, se requiere el adaptador anteriormente mencionado.

Miguel Cabezuelo 3D

Otras características

Tubo de imagen
17" (diagonal) plano, rectangular
16" (diagonal del área visible)
Tamaño de punto de 0,28 mm.

Resolución y frecuencia de refresco
640 x 480 píxeles a 85 Hz
800 x 600 píxeles a 85 Hz
832 x 624 píxeles a 75 Hz
1.024 x 768 píxeles a 85 Hz
1.152 x 870 píxeles a 75 Hz
1.280 x 1.024 píxeles a 60 Hz

Frecuencias de barrido
36 KHz a 69 KHz (horizontal)
48 Hz a 160 Hz (vertical)

Controles del usuario
- Interruptor de encendido
- Desmagnetización automática al encender
- Desmagnetización manual por control en pantalla
- Ajuste/selección del gestor en pantalla
- Brillo
- Contraste
- Controles de imagen adicionales mediante la gestión en pantalla

Homologaciones
- FCC Class B
- VDE Class B, FTZ
- CISPR 22 Class B (EN55022)
- TCO 95 (mundial)
- Aprobación CE
- PFC (mundial) TCO modificado; MPR II magnéticos y eléctricos; VLF magnéticos, eléctricos y electrostáticos (todos los formatos de monitor)
- DPMS
- EPA Energy Star

Dimensiones y peso
Alto: 43,1 cm
Ancho: 41,2 cm
Largo: 42,8 cm
Peso: 17,0 kg

Power Macintosh 5500/225

El Power Macintosh 5500 de Apple es el ordenador Macintosh más potente, ampliable y con más funciones diseñado específicamente para la familia. La impresionante velocidad del procesador PowerPC, combinada con la aceleración por hardware de gráficos y vídeo, permite que los usuarios realicen tareas que antes sólo eran posibles en costosas estaciones de trabajo profesionales. Tanto si queremos establecer una videoconferencia por Internet, entretenernos con alguno de los juegos más populares, crear un vídeo con una banda sonora propia o navegar por la Red, el Power Macintosh 5500 hará que la experiencia resulte sencilla, dinámica y divertida.

Para poder adaptarse a la vida en familia, el Power Macintosh 5500 incorpora un diseño modular. Es posible inclinarlo y girarlo, combinando la sencillez del diseño modular sin perder el confort. Pero los ingenieros de Apple no se contentaron con esto: han adaptado el sistema por completo para manipular la riqueza de medios que hoy se encuentran en Internet y en los CD-ROM, sin sacrificar la "sólida" estabilidad y la facilidad de uso.

El avanzado subsistema de gráficos y vídeo acelera la manipulación de complejos documentos 2D mediante aceleración QuickDraw. Permite concentrarse en un juego 3D y en el modelado 3D interactivo con la aceleración QuickDraw 3D y puede reproducir películas a pantalla completa mediante QuickTime y descodificación MPEG. Asimismo, conecta el sonido estéreo con calidad CD de sus altavoces integrados y el color de 24 bits del monitor de 15" integrado, y con ellos podremos realizar prácticamente cualquier cosa que queramos hacer. Además, para poder disfrutarlo desde el principio, el Power Macintosh 5500 incluye una valiosa colección de software cuidadosamente seleccionada.

Miguel Cabezuelo 3D

Características

- Procesador RISC PowerPC 603e a 225 MHz.
- Incorpora 256 KB de caché nivel 2.
- 32 MB de EDO RAM, ampliables a 128 MB mediante 2 ranuras DIMM.
- Disco duro IDE de 2 GB.
- Unidad de disco interna Apple SuperDrive (1,4 MB). Lee, escribe y formatea discos Macintosh, Windows, MS-DOS, OS/2 y ProDOS.
- Unidad de CD-ROM 24x (máximo).

Multimedia
- Aceleración por hardware de gráficos 2D y 3D, captura de vídeo y reproducción de películas QuickTime y MPEG-1.
- Chip acelerador de gráficos y multimedia ATI 3D Rage II de 64 bits, para capturas de vídeo más rápidas y continuas; aceleración de gráficos QuickDraw (2D) y QuickDraw 3D y reproducción de películas QuickTime y MPEG-1.
- Admite color de 24 bits en el monitor de 15" incorporado (medida diagonal).
- Unidad de CD-ROM 24x (máximo) de carga por bandeja.
- Sonido estéreo con calidad CD de 16 bits, con sonido envolvente SRS.
- Puertos de E/S estéreo, calidad CD, 16 bits.
- Realce por sonido envolvente SRS 3-D.
- Altavoces estéreo integrados.

Acceso a Internet
- Módem interno datos/voz/fax que admite transferencia de datos de hasta 33,6 Kbit/s.
- Software preinstalado para acelerar y simplificar la navegación por la web.
- Envío/recepción de e-mail, transferencia de archivos e incluso la creación de un servidor web propio.

Expansión y flexibilidad
- Diseño modular con una ranura de expansión PCI estándar.
- Admite opciones como el sistema de edición de vídeo Avid Cinema y el Apple TV/Radio FM System.
- Instalación y ejecución instantánea de las tarjetas de expansión, dispositivos de almacenamiento, escáneres, impresoras, cámaras digitales y otros accesorios.

Software incluido
- Mac OS 8.
- Software Guía Apple con ayuda paso a paso sobre funciones básicas del sistema.
- Software multimedia QuickTime.
- Software Macintosh PC Exchange.
- Incluye una amplia colección de software, cuidadosamente escogida para responder a las necesidades de la familia.

Dimensiones y peso
Alto: 44,5 cm
Ancho: 38,3 cm
Largo: 40,6cm
Peso: 21 kg



Una nueva realidad en

La verdad es que este artículo hace un par de años no hubiera sido posible. Es increíble la cantidad de herramientas que han aparecido para abordar el modelado orgánico. Sólo hay que mirar un poco hacia atrás para darse cuenta que, salvo excepciones, el modelado de formas orgánicas era algo al alcance de muy pocos.

Esa época pasó a mejor vida. Ahora es fácil encontrar miles de modelos por la red, creados en apenas unas horas, que antes sólo podíamos imaginar de estaciones SGI o de auténticos profesionales. La evolución que ha sufrido el paquete 3D Studio en el aspecto orgánico ha sido enorme. De la primera versión de 3DS hasta el novedoso Max 2.0, con Nurbs incluidas, han pasado muchos años. Es lógico pensar que en todos estos años algo ha tenido que cambiar, y, en efecto, algo ha cambiado y de qué manera.

Cualquiera que todavía diga que 3DS MAX no tiene capacidades de modelar es porque o no ha trabajado lo suficiente con él o, simplemente, habla por hablar.



Imagen 1. Modelo creado bajo 3DStudio con Metareyes 2.0.

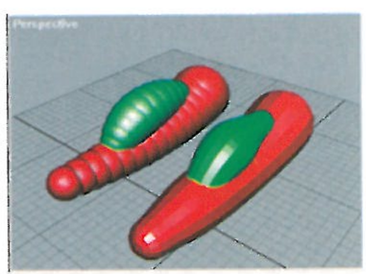


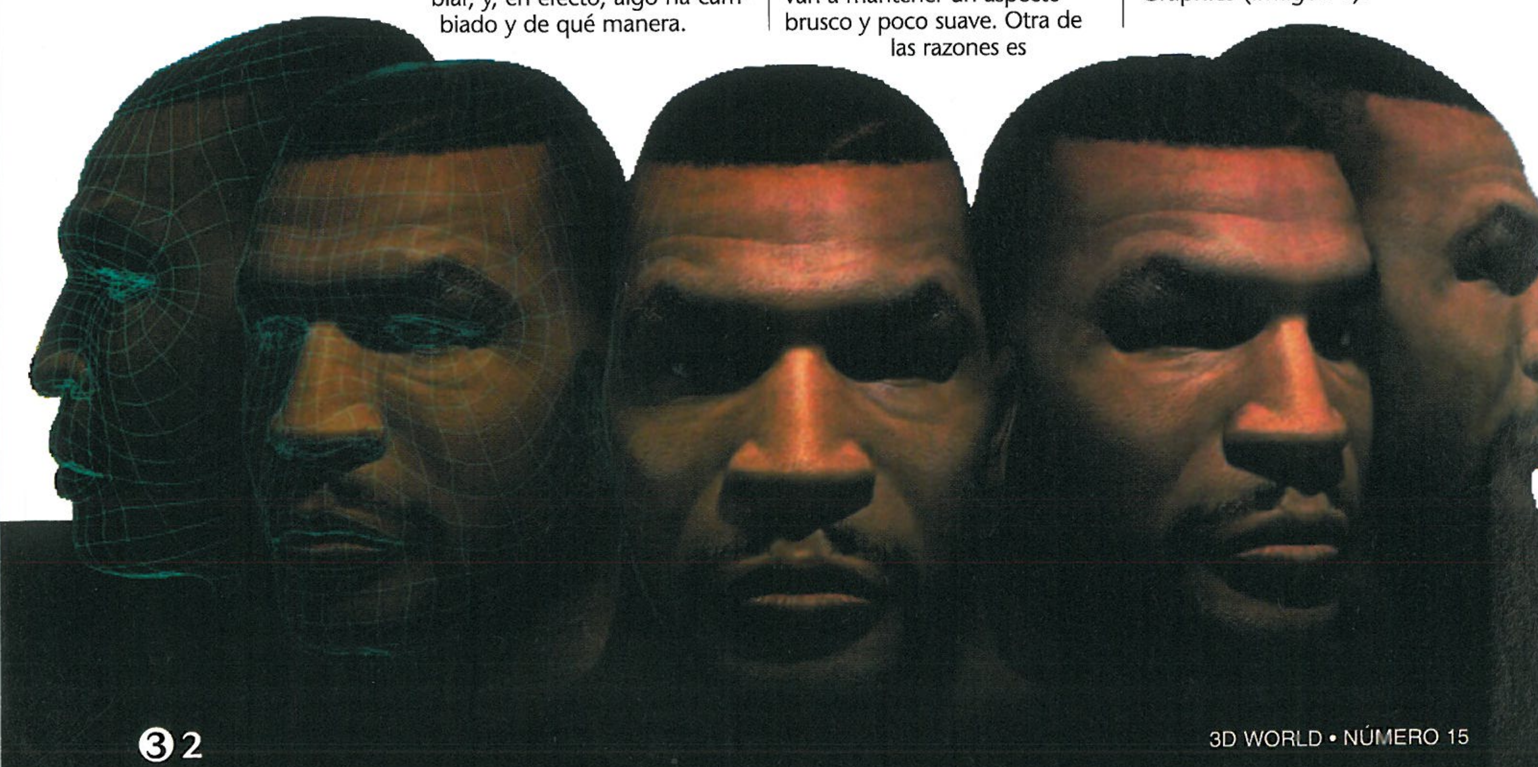
Imagen 2. La diferencia de primitivas es considerable entre las dos versiones.

Cualquiera que todavía diga que 3DS MAX no tiene capacidades de modelar es porque o no ha trabajado lo suficiente con él o, simplemente, habla por hablar. Para todos aquellos que sigan dejando a un lado al Max a la hora de la creación, va dedicado este artículo.

Apariencia orgánica

¿Por qué es tan difícil dar apariencia orgánica a un modelo? Hay varias razones. En primer lugar, nos tenemos que fijar con qué estamos trabajando. Hay que recordar que manipulamos polígonos y que siempre van a mantener un aspecto brusco y poco suave. Otra de las razones es que, al modelar, vamos a obtener resultados bastante perfectos debido a simetrías y demás. Esto hay que evitarlo en la medida de lo posible para conseguir salir de ese perfeccionismo que tan poco abunda en la naturaleza. ¿Soluciones? Muchas. Vamos a verlas:

- **Metareyes:** No podíamos empezar por otro que no fuera el plug-in Metareyes. Fue la primera herramienta que haría posible la creación de formas orgánicas en un PC sin tener que irse a las, por aquel entonces, super-poderosas Silicon Graphics (imagen 1).



tus personajes



Imagen 3.

Era el principio de lo que se iba a convertir en el estándar del modelado orgánico. De las miles de esferas del Metareyes 2.0 para 3DStudio se pasó a unos cuantos músculos del Metareyes 3.0 (imagen 2). Se había conseguido más facilidad y rapidez, aparte de economizar recursos dando unos resultados sorprendentes. No es de extrañar el éxito obtenido.

En Metareyes 3.0, lo único que debe hacer el modelador es ir situando los distintos *metamuscles* en una de las vistas, para luego pasar a darles la tercera dimensión y conseguir modelos asombrosos (imagen 3). Para los que aún desconozcan el funcionamiento de las *metaballs* y de los *metamuscles*, decir que es como tener mercurio dentro de la pantalla. Esos músculos se atraerán o repelerán unos a otros según el color que les asignemos.

Como se aprecia, las ventajas son muchas pero también tiene sus defectos. En primer lugar, el control sobre la densidad de la malla. Es una de sus peores bazas, pues debemos andarnos con cuidado si no queremos encontrarnos con un simple modelo que se coma los Mb de Ram como si fueran simples bytes.

Otro de los problemas es que el número de vértices se reparte por igual por todo el modelo teniendo, así, el mismo número en zonas grandes, donde no hace falta mucho detalle, que en zonas pequeñas donde el detalle es fundamental (imagen 4). Imaginar un dragón con un cuerpo enorme lleno de vértices mientras que la cara, que es donde más detalle hay, se tiene que conformar con unos pocos. El reparto de tantos vértices por centímetro cuadrado no es la mejor opción en la mayoría de los casos.

Otro gran problema son las superficies que no necesitan formas esféricas para su construcción. Metareyes 3.0 no puede hacer nada frente a eso. Tomad el ejemplo de un delfín. No es posible hacer la malla de una sola vez. El cuerpo no sería ningún problema, pero las aletas casi planas es un obstáculo insalvable para las formas esféricas. De todas maneras, los chicos de REM, que están a todas, ya tienen preparada la versión 4.0 de Metareyes que resolverá éstas y otras pequeñas deficiencias (imagen 5).

- **Surface Tools:** Uno de los grandes desconocidos para Max. Distribuido por Digimation y creado por Peter Watje, tenemos en Surface Tools uno de los plug-ins más sencillos y más sorprendentes que hay por el momento. Si queremos obtener modelos extremadamente realistas, este plug-in es el indicado. Su funcionamiento os parecerá extremadamente rudimentario pero los resultados obtenidos no lo son tanto. Se basa en el modelado mediante *splines* y *patches*. El modelador va creando los vértices y líneas en que se va a basar su modelo. Es cómo ir creando el molde que luego se transformará en la escultura (imagen 6).

Sólo nos tenemos que encargar de crear unas cuantas líneas de nuestro modelo, pues Surface Tools hará el resto. De lo único que hay que preocuparse es de que esas líneas (*splines*) que creemos se unan para crear secciones con tres o cuatro aristas (imagen 7). Esto se explica a la hora de ver en acción al plug-in. Éste rellenará esas secciones con *patches* del tipo "tri" o "cuad" dando gran detalle al modelo. Su funcionamiento es muy similar al modificador MeshSmooth sólo que, al trabajar con *splines* y *patches*, podemos ir a cualquier nivel de la pila de modificadores y tener un control total sobre el objeto que estemos creando.



Imagen 4. Podemos apreciar claramente como el cuerpo está ampliamente detallado (demasiado) mientras que la cara está completamente deformada.



Imagen 5.



Surface Tools aplicado

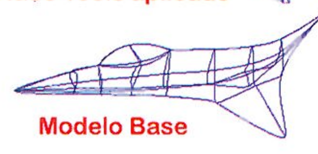


Imagen 6.

El resultado final es sorprendente pero, cómo no, este plug-in también tiene algún que otro defecto. Al tener que crear todas las líneas y vértices de nuestro modelo, se hace especialmente pesado y trabajoso. También se llega a un momento en el que te encuentras entre una maraña de vértices y líneas donde es un auténtico suplicio seguir trabajando.

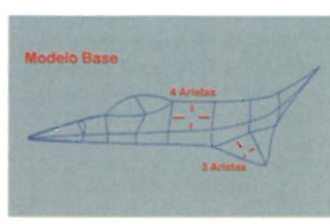


Imagen 7.

Con estos dos plug-ins no habrá modelo orgánico que se nos resista. Os recomendamos utilizar Metareyes para crear cuerpos y superficies poco detalladas, sin embargo si vais

Otro gran problema son las superficies que no necesitan formas esféricas para su construcción

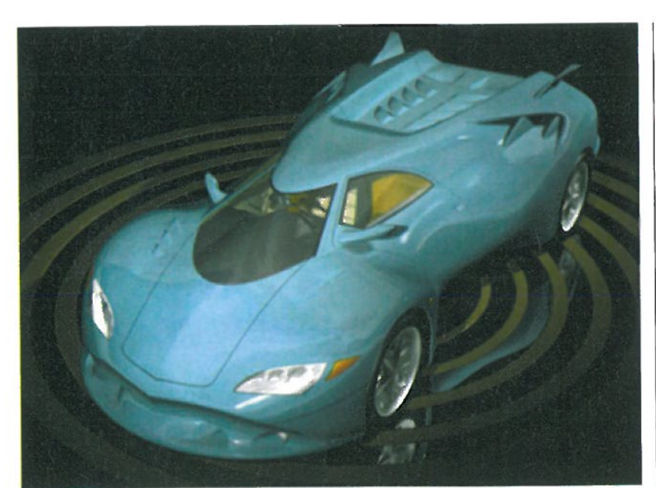


Imagen 8. La capacidad de modelado de Surface Tools queda patente con este modelo de Leonardo Paredes. El detalle que se alcanza es increíble.

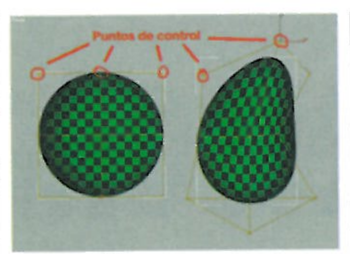


Imagen 9.



Imagen 10.

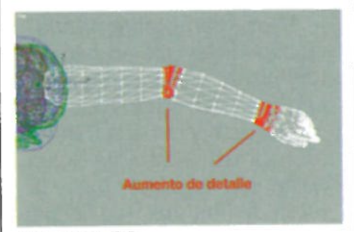


Imagen 11.

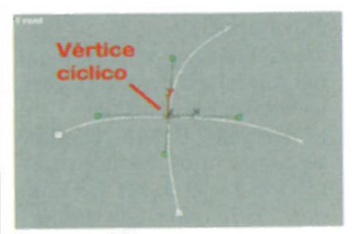


Imagen 12.

a modelar caras humanas u otras mallas, donde se necesite tener un control más detallado, la elección va hacia Surface Tools (imagen 8).

• **Otras opciones:** Existen otros modificadores que también van a venir muy bien en el modelado orgánico: MeshSmooth, FFD, Relax... Con ellos lo que conseguimos será suavizar y modificar al objeto en aquellas partes donde lo necesite. Como se ha dicho antes, los plug-ins Metareyes y Surface Tools crean mallas homogéneas con el mismo nivel de detalle en todas las partes del objeto. Es aquí donde entran los modificadores nombrados. Los podemos aplicar a nivel sub-objeto y aumentar la densidad de vértices. Esto lo conseguimos con MeshSmooth. Tendremos más vértices allí donde queramos, para poder manipularlos y darles forma. Es el turno del modificador FFD (*Free Form Deformation*). Con él aplicamos una especie de jaula que encierra al objeto o parte seleccionada del mismo. Esta jaula estará formada por unos puntos de control que actuarán como imanes. Si tiramos de los puntos, estiraremos la zona adyacente del objeto de forma suave (imagen 9). Esa suavidad es la que le hace tan indicado para la modificación de objetos orgánicos.

Una vez que tenemos el armazón y las piezas detalladas, debemos asignar Optimize y Relax al armazón para eliminar el excesivo número de vértices generados y suavizar el modelo, respectivamente



Pero si lo que queremos es quitar densidad a la malla, suprimir vértices, debemos actuar de otro modo. Primero tenemos que aplicar el modificador *Optimize* y ajustar los niveles para que la malla resultante sea de nuestro agrado. Si apreciamos una pérdida excesiva de suavidad al eliminar vértices, debemos aplicar el modificador *Relax*. Como su nombre indica, lo que hará será relajar la malla para que no haya zonas bruscas que delaten nuestro modelo (imagen 10).

Ya, por último, os indicaremos la manera, digamos profesional, de crear modelos para animación. En primer lugar usaremos Metareyes para obtener la base del que será nuestro modelo. Éste nos saldrá con excesivo detalle debido a la gran cantidad de vértices que Metareyes genera. Utilizaremos Surface Tools para los elementos que vayan a ser más utilizados en la animación y que requieran de mayor calidad. Por ejemplo, las manos, la cabeza y los pies si el modelo es humano o animal. Una vez que tenemos el armazón y las piezas detalladas, debemos asignar *Optimize* y *Relax* al armazón para eliminar el excesivo número de vértices generados y suavizar el modelo, respectivamente. Por último, aplicaremos *Mesh Smooth* a nivel de sub-objeto en las articulaciones. Con ello conseguiremos aumentar la densidad en zonas como codos, hombros, muñecas, etc., para que, al aplicar un sistema basado en huesos, las articulaciones, que son las que más sufren, no se resquebrajen y hagan los giros y movimientos con suavidad (imagen 11).

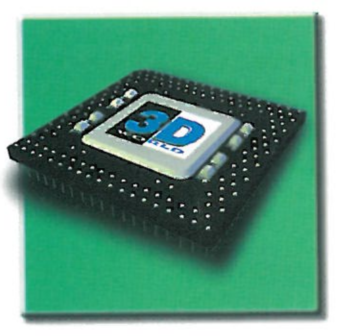
Bueno, pues ya tenemos un modelo creado a la perfección dispuesto a ser animado. No desesperéis al ver que os lleva bastante tiempo, los resultados merecerán la pena.

Ya sabéis, si tenéis alguna duda o sugerencia: REYMAD@santandersupernet.com

Miguel Angel Díaz Martín

Utilización de *SURFACE TOOLS*

El funcionamiento es muy sencillo. Se trata de ir creando los vértices por los que pasarán las líneas que va a tener nuestro modelo. Para ello utilizamos el modificador Edit Spline 2 que nos permitirá crear vértices cíclicos. Este tipo de vértice nos posibilita cruzar dos líneas que tengan direcciones distintas, es decir, un vértice del que pueden salir más de dos líneas (ver imagen 12). Lo ideal sería tener dos fotografías del modelo a realizar. Crearíamos unas cuantas líneas de contorno y luego esos contornos los rellenaríamos con otra serie de líneas que servirán de guía para que *Surface Tools* coloque los *patches* que darán el aspecto final al modelo (ver imagen 13).



Portátiles WorkStation

Toshiba Tecra 750 DVD vs IBM ThinkPad 770

Autor: Enrique Urbaneja

Si eres uno de los pocos, pero cada vez más, privilegiados que piensan adquirir un portátil Workstation y no sabes por donde van los tiros actualmente, esta comparativa te muestra dos titánicos, en lo que a capacidad de procesamiento y prestaciones se refiere, que podrían ridiculizar a muchos ordenadores de sobremesa actuales.

Varios son los factores a tener en cuenta cuando nos disponemos a comprar un ordenador portátil. Sin embargo, el problema no reside en esta variedad, sino en que la mayoría de estas características nos son desconocidas porque nuestros ordenadores de sobremesa no las poseen. Hablamos de características como la autonomía, el tipo de pantalla o el replicador de puertos, que serán comentadas brevemente a continuación y que pueden hacer subir el precio de nuestro portátil en un abrir y cerrar de ojos.

Comenzando por el tipo de pantalla, actualmente existen dos tecnologías, DSTN y TTF, que será lo primero que tendremos que tener en cuenta cuando vayamos a comprar un equipo portátil. La diferencia principal estriba, dejando a un lado la tecnología empleada, en que el primer tipo necesita que nuestra posición frente a la pantalla sea completamente perpendicular, por lo que un segundo usuario que intentase ver algo mientras nosotros trabajásemos lo tendría realmente difícil. Este problema no existe hasta cierto punto con las pantallas TTF, ya que el ángulo de visión abarca un rango de unos 60°, más que suficiente para que tres o cuatro personas sentadas a nuestro lado puedan ver perfectamente la información.

Por supuesto, el último tipo de pantallas son mucho más caras que las DSTN con diferencias de hasta 120.000 pesetas.

En cuanto a la autonomía de nuestro equipo, depende directamente del tipo de batería que utilice, aspecto a estudiar ya que existen diferentes tipos con diversas características de peso, duración, ergonomía, etc... En la actualidad existe cierta declinación hacia las baterías de iones de litio, debido a sus características de peso y capacidad.

El resto de opciones se comentarán a continuación, viendo *in situ* las prestaciones de cada equipo.

TOSHIBA VS IBM

Como el lector se habrá percatado, los dos fabricantes de los equipos de la comparativa no son unos principiantes en el mundo de la informática ni en el de los portátiles.

Toshiba Tecra 750 DVD se muestra como uno de los portátiles Workstation del mercado por excelencia: procesador Pentium MMX a 233 Mhz con 64 MB de Ram y 5 GB de disco duro.

Sin duda alguna, toda una estación gráfica portátil con pantalla TTF de 13,3 pulgadas y su tarjeta gráfica con aceleradores 2D y 3D por hardware de la marca Virge/MX con 4 MB RAM.

Por su parte, el ThinkPad 770 de IBM procesa con un procesador Pentium 233 Mhz MMX con 32 MB RAM y 4,3 Gb de disco duro, en donde se pueden apreciar las primeras diferencias entre equipos, a primera vista, ya que un punto a favor de ThinkPad es su capacidad de ampliación hasta 256 Mb de RAM, mientras que Tecra sólo podría ser ampliado hasta 160 MB RAM.





CUADRO COMPARATIVO DE CARACTERÍSTICAS

	IBM ThinkPad 770	Toshiba Tecra 750DVD
Dimensiones: (cm)		
Ancho	31	30,5
Alto	6	5,5
Fondo	25	24,5
Peso total	3,6	3,6
Tiempo de autonomia (h)	3	4
Tiempo de carga aprox. (h)	2	2,37
Bus gráficos	PCI	PCI
Res. máxima	1.280x1024	1.600x1200
Nº colores máximos	64K	16M
Memoria vídeo instalada/máxima	2/2	4/4
Caché	512	512
Tipo de memoria	SDRAM	EDO RAM
Nº zócalos	2 SODIM	2 SODIM
CD-ROM	20x	20x
Módem	56 Kbps	33 Kbps
Precio aproximado	1.350.000 ptas.	1.200.000 ptas.

OTRAS CARACTERÍSTICAS

	IBM ThinkPad 770	Toshiba Tecra 750DVD
Nº teclas	86	86
Nº teclas especiales	7	9
Huecos Multifunción	2	1
Posibilidad abatir teclado	No	No
Tipo panel informativo	LCD	LED
Soporte PC Card III	Sí	Sí
MPEG hardware	Sí	Sí
Escritorio virtual	Sí	Sí
El equipo se apaga con interruptor	Sí	No
Ranuras PC Card II	2	2
Tipo apuntador	TrackPoint	TrackPoint

Y éste es uno de los hechos que no nos debe confundir a la hora de adquirir uno de estos ordenadores: la configuración inicial puede ser muy sabrosa, pero nunca se debe olvidar echar un vistazo a las posibilidades de ampliación del equipo.

Otra diferencia en cuanto a la memoria RAM entre estos dos portátiles es la tecnología. La memoria Ram del Tecra es tecnología EDO, mientras que la del ThinkPad es SDRAM, no muy relevante pero que hay que conocer.

Al igual que ocurre en el Tecra, la pantalla del portátil de IBM es TTF y casi de las mismas proporciones, 13,1 pulgadas, con un sistema de vídeo movido por un acelerador gráfico de la marca Trident, el Cyber 9397 con 2 Mb de memoria RAM y que también cuenta con aceleración 2D y 3D, aceleración 3D que llega casi a doblar en rendimiento a las prestaciones del Tecra de Toshiba, característica a tener en cuenta para aquellos que quieran sacar partido a su modelador 3D.

En cuanto a los dispositivos como CD-ROM o disquetera, ambos presentan el inconveniente de tener que poner o uno u otro, aunque, de esta forma, se gana en dimensiones, sin olvidar que los dos ordenadores tienen la posibilidad de cambiar el dispositivo en caliente.

Pocas cosas quedan ya por comentar, si bien el tema de puertos y conexiones no ha sido tocado, en donde el ThinkPad posee puertos para las conexiones normales como puerto serie, paralelo, una salida para monitor externo, entradas y salidas para audio, mientras que Tecra, además de éstos, incorpora un puerto de conexión por infrarrojos tanto en la parte posterior como en la frontal, todo un detalle de calidad.

Para mayor información, se pueden consultar las siguientes direcciones:

Toshiba
Tel: (902) 21-11-21
http://www.toshiba.com

IBM
Tel: (900) 100-400
http://www.ibm.es

A DESTACAR

Toshiba Tecra 750 DVD tiene una característica envidiable, y ésta es la que protagoniza su equipo de vídeo, ya que dispone de una cámara que se sitúa en el lado derecho de la pantalla y que hará las delicias de aquellos que deseen utilizar videoconferencia, por lo que cuenta además con un módem interno a 33.600 bps y software para videoconferencia: todo un lujo.

IBM ThinkPad 770 tiene también su característica particular: lleva por hardware un descompresor de MPEG-2 y Dolby AC-3 permitiendo, de esta forma, visualizar películas en DVD, cuyo lector se instalaría en el hueco donde se aloja el CD-ROM o la disquetera.



Claves de la Infografia Profesional

La animación de personajes (V)
Autor: Jesús Nuevo España

Nivel: Medio

Este mes vamos a hablar sobre la animación antropomórfica, veremos cómo animar personajes de forma estática y personajes maleables, y terminaremos con algunos consejos y trucos muy interesantes para resolver algunas situaciones delicadas.

Gracias a la Infografía, la Comunicación Audiovisual (especialmente el Cine) ha podido llevar a la pantalla a animales que se extinguieron hace miles de años, o lugares que se hallan a miles de millones de kilómetros de la Tierra (si es que existen realmente), o actores que llevaban muertos varios años. Todo con un grado de realismo tal, que el espectador cree que cuanto ve ha sido filmado con una cámara, es decir, que no existe manipulación alguna. Esto es así, gracias a que las imágenes poseen ese *don* que además de verosímiles las hace creíbles: el hiperrealismo.

Si hay algo que diferencia a la Animación Tradicional (Dibujo Animado) y a la Infografía es precisamente eso. Los animadores siempre han buscado imitar lo que sucede en el mundo real. Mediante una depurada utilización de unos principios básicos, en combinación con determinados convencionalismos del mundo de la comunicación gráfica, se consiguió que los trazos de colores cobrasen vida cada vez con mayor similitud y su analogía fue evolucionando cada vez a cotas mayores. Pero a pesar de todo, el Dibujo Animado nunca podría dejar de ser eso: Dibujo. La Infografía viene a llenar ese vacío gracias a sus enormes posibilidades (que además siguen en constante crecimiento). Se logran resultados tan espectaculares, que en muchos casos resulta imposible saber qué es real y qué es imagen de síntesis.

Hoy en día a nadie le extraña contemplar c,omo en un *Spot* publicitario un paquete de cereales comienza a caminar, o cómo un grupo de caramelos de colores saltan por toda la casa. Tanto han evolucionado las técnicas de tratamiento digital de la imagen que nos resulta absolutamente "normal" que esto suceda. Pero, ¿cómo se puede hacer que algo tan radicalmente distinto de un cuerpo humano, como pueda ser un paquete de cereales, cobre vida y se mueva, como si de una persona se tratase? Muy sencillo, aplicando técnicas avanzadas de animación, o lo que es lo mismo, con Animación Antropomórfica.

LOS PERSONAJES

Atendiendo a su morfología, podemos encontrar multitud de personajes, pero ahora vamos a centrarnos en aquellos que simulan una apariencia y un comportamiento humano, es decir, los llamados personajes antropomorfos. Al hablar de los personajes antropomorfos, algunos autores los agrupan en dos categorías: *personajes de forma estática* y *personajes maleables*.

Los *personajes de forma estática* se mueven de la misma forma que lo harían en la realidad. Un claro ejemplo sería *Luxo Jr.*, el personaje de Pixar. Esta lámpara-flexo se compone sólo de tres articulaciones, de manera que evoluciona por el espacio a partir de las diferentes flexiones que le permite su limitada anatomía. Aunque a priori podría parecer que sólo tres articulaciones son pocas para conseguir un movimiento natural, lo cierto es que con una correcta utilización de la medida del tiempo en el control de dichos movimientos se consigue que parezca completamente real. Su forma permanece inalterable. Se mueve, se flexiona, pero no se altera su morfología. Este tipo de personajes son idóneos para animar juguetes mecánicos (tal y como quedó demostrado en la última producción de la citada *Pixar* y la factoría *Disney*, la maravillosa *Toy Story*). Las herramientas empleadas para la animación de este tipo de personajes son las ya habituales *Cinemáticas Inversa/Directa*, *Bones* (huesos), *Generadores de Huellas*, etc.

Los personajes maleables son aquellos que, no teniendo una apariencia antropomorfa, sí tienden a comportarse como los seres humanos. Estos personajes se animan utilizando lo que llamamos el método de modelado. Este método otorga al personaje unas cualidades que en la vida real nunca poseería, lo que le permite estirarse o aplastarse de una forma anormal. Un bote de champú o una lata de cerveza no poseen articulaciones, de manera que, para que puedan caminar como un ser humano, hemos de definir nosotros esas articulaciones (mediante el método de modelado).

Muchos animadores, antes de dar vida a un objeto, realizan un estudio para determinar a qué tipo de animal se parece más. Así nos encontramos que las mesas y las sillas se adaptan bastante bien a la anatomía de un perro. Las cajas suelen adaptarse bien a movimientos bípedos, de manera que las esquinas inferiores se comporten como si fuesen los pies y las superiores como si de los hombros se tratase.

El tamaño va a ser otro factor que va a determinar en gran medida el movimiento de nuestro personaje. Un objeto decorativo (jarrón, cuadro, etc.) tenderá a moverse mucho más rápido que un bloque de pisos o que un puente.

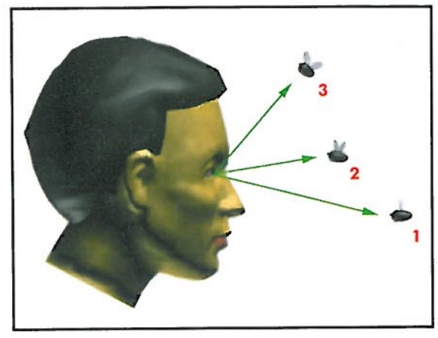
Otro problema que nos encontramos se deriva de la decisión de diseñar o no una cara para el personaje. Como ya hemos comentado, el personaje podrá expresar muchas emociones con la postura del cuerpo. Pero si el personaje tiene que hablar entonces sí se hace imprescindible una cara. Tendremos que ver de qué manera podemos adaptar partes huecas para que simulen ser la boca, o salientes que simulen ser los ojos, etc.

CONTROLANDO LA MEDIDA DEL TIEMPO

Antes de comenzar el proceso de animación conviene hacer distintas pruebas, con el objeto de determinar las dificultades que presenta el personaje para mostrar determinadas expresiones. Si el personaje carece de rostro, todos los estados de ánimo dependerán exclusivamente de la postura y del control de la medida del tiempo. Este tipo de animaciones requiere que éstas (las posturas) sean más exageradas de lo habitual, es decir, que posean un significado añadido, del cual se va a desprender el sentimiento o el estado de ánimo implícito. Esto hace que sean animaciones muy divertidas, que suelen contar con el beneplácito del público.

El secreto está en hacer un correcto empleo de la medida del tiempo, gracias a algunas de las técnicas que vimos en números anteriores. Concretamente podemos destacar cinco: *Squash and Stretch, Anticipation, Exaggeration* y *Follow Trhought.* Habremos de tener mucho cuidado, dado que en este tipo de animaciones el más mínimo error en la sincronización del tiempo a los ojos del espectador aparecerá con desmesurada gravedad.

Cuando animamos *personajes de forma estática,* rápidamente nos damos cuenta de que los únicos puntos móviles son las articulaciones. Las acciones de *Squash and Stretch* quedan prohibidas, ya que no podemos cambiar la forma del personaje. Tendremos que crear su personalidad a partir de *Anticipations, Exaggerations* y *Follow Trhought.* Por supuesto, que tendremos que establecer correspondencias anatómicas respecto del cuerpo humano, de manera que si el objeto es un brazo mecánico pues se comporte precisamente como eso: como un brazo. O por ejemplo, si hemos de animar una guitarra, el mástil será el cuello, la caja de resonancia se comportará como el cuerpo (teniendo que desplazarse a partir de *Squash and Stretch*) y el clavijero como la cabeza.



LA HERRAMIENTA "MIRAR A" PERMITE SEGUIR CON LA MIRADA A OTRO OBJETO.

Cuando animamos *personajes maleables* podemos hacer que cambien de forma conforme se mueven, de manera completamente libre, lo que puede hacer que nuestras animaciones alcancen un paroxismo algo surrealista. Las herramientas varían de un paquete de software a otro, por lo que tendremos que procurar sacarle todo el jugo a las que tengamos. A veces, como dice George Maestri, la clave para una animación genial está en crear una solución original e ingeniosa basándonos en recursos limitados.

DEFORMACIONES SIMPLES

Muchas veces, las técnicas más simples suelen dar los resultados más espectaculares. Así sólo con escalados o cambios de tamaño, dobleces, giros o torsiones y husos podemos conseguir que una simple *Box* (caja cúbica) cobre vida. Si vamos a hacer que nuestra caja se doble por la parte media, a modo de saludo, tendremos que modelarla con varias subdivisiones (Figura 2).

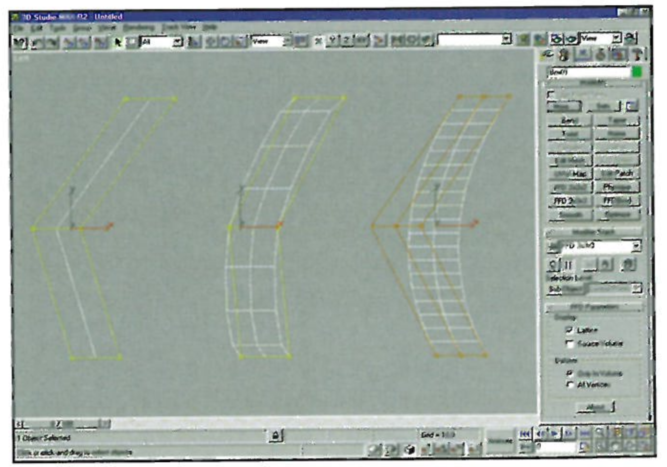
Con una simple torsión creamos la ilusión de que el personaje posee caderas y hombros, de manera que al caminar giran hacia delante y hacia atrás. Incluso podemos simular giros de cabeza o hacer que el personaje mire por encima del hombro girando sólo la parte superior.

Con los husos podemos simular el peso del personaje, haciendo que se comprima al tomar contacto con el suelo. También podemos hacer que el personaje saque pecho o por el contrario que se desinfle.

AQUÍ PODEMOS CONTEMPLAR UN ENREJADO DE DEFORMACIÓN.



CUANTAS MÁS SUBDIVISIONES, MÁS SUAVE ES LA DOBLEZ REALIZADA.



Si empleamos correctamente todas estas herramientas de deformación conseguiremos resultados sorprendentes. Lo importante es saber qué deformación debemos emplear para representar cada gesto, cada expresión. No siempre son necesarias sofisticadas herramientas de animación para representar movimientos visualmente atractivos.

Los personajes antropomorfos pueden ser de forma estática o maleables

DEFORMACIONES COMPLEJAS

Existen otras herramientas de deformación no tan simples, que pueden servirnos para animar *personajes maleables*. Entre ellas podemos destacar tres: los enrejados de deformación, las curvas flexibles y los huesos.

Los enrejados están formados por multitud de celdillas, con puntos de control en sus esquinas, que permiten que las modifiquemos libremente. Los usuarios del 3DS MAX estarán familiarizados con el empleo de este tipo de herramientas, que aquí se conocen con el nombre de FFD (*Free Form Deformation*). Y hemos de advertir que, a pesar de su sencillez, las posibilidades que permite, tanto para modelado como para animación, son increíbles. Con un mínimo esfuerzo podemos crear deformaciones maleables interesantes. Lo habitual es que se apliquen enrejados cúbicos, aunque ya hay algunos paquetes de software (como 3D Studio MAX II) que permiten enrejados cilíndricos o de otras geometrías. Para animar esa caja de cereales o un paquete de tabaco es la herramienta ideal. Sólo necesitamos que nuestro modelo tenga suficientes subdivisiones (ya lo habíamos comentado antes) de manera que las torsiones, giros y escalados sean soportados óptimamente y no nos aparezcan tensiones o distorsiones no deseadas. Así, moviendo uno o dos puntos de control de nuestro enrejado, podemos cambiar completamente la forma docenas de vértices del objeto.

También podemos hacer que nuestro modelo se doble a lo largo de una curva flexible. Así, cuando manipulamos la curva flexible que controla al personaje, éste reacciona siguiendo su movimiento. Esta técnica, no es demasiado habitual, pero es ideal para animar objetos con forma de serpiente.

Mucho más frecuente resulta la utilización de huesos (Bones) en la animación de personajes. Gracias a ellos, objetos que resultan prácticamente imposibles de animar pueden llegar a cobrar vida. Lo más difícil es determinar el número de huesos y la ubicación de los mismos. Además permiten aplicar CI (Cinemática Inversa), lo que facilita mucho el proceso de animación, permitiendo anclar los pies durante los ciclos de animación o interacciones reales al establecer contacto con otros objetos (a partir de jerarquías). Sin duda, una herramienta muy potente que nos permitirá aumenta más aún la calidad de nuestras animaciones.

CONSEJOS Y TRUCOS

Hasta ahora hemos hablado de situaciones aisladas, pero lo cierto es que el mundo animado es generalmente mucho más complicado. Los personajes no están solos, ni sus acciones son tan aisladas. Y es precisamente ahí donde suelen surgir los problemas, cuando hacemos que los personajes interactúen más los unos con los otros y con el ambiente en que se desenvuelven.

Existen además multitud de situaciones que, aunque a priori pueden parecernos sencillas de resolver, a la hora de realizar la animación nos damos cuenta de que presentan serias dificultades. Eso en muchas ocasiones puede ocasionarnos graves retrasos e incluso puede paralizar por completo el ritmo de producción. Por ello quiero que veamos algunos trucos y consejos para resolver con éxito esas situaciones tan particulares.

1. Levantar objetos: un problema bastante frecuente al que se enfrentan los infografistas a la hora de realizar sus animaciones es conseguir que un personaje levante un objeto y que éste se mantenga en sus manos de una forma natural, es decir, que siga a la mano en sus movimientos. Esto sería sencillo de resolver utilizando una jerarquía variable entre la mano y la copa, de manera que hasta el instante en que se produce el contacto ambos estén al mismo nivel y justo cuando la mano coja la copa, ésta pase a depender de la mano. Pero lo cierto es que, salvo honrosas excepciones como el programa *Lock and Key* para *LightWave*, la mayoría de los paquetes no permiten este tipo de cosas.

Un modo bastante efectivo para resolver esta situación consiste en segmentar la escena. De este modo podemos fragmentar la acción en tres planos distintos, utilizando distintas jerarquías en cada uno de ellos: en el primer plano (en plano medio) se mostraría al personaje acercando su mano a la copa; en el segundo (en plano detalle) se vería la mano levantando la copa (en este plano la copa ya depende jerárquicamente de la mano); en el tercero el personaje (nuevamente en plano medio) levanta la copa y se la lleva a la boca.

Otra solución bastante inteligente consistiría en ocultar y mostrar dos objetos idénticos. ¿Cómo? Muy sencillo, el secreto está en disponer de dos copas idénticas. Bien, pues lo que hacemos es colocar una en la mesa y otra en la propia mano del personaje. La de la mesa está desconectada de él, mientras que la otra depende jerárquicamente de la mano. El truco consiste en que la segunda estará oculta. ¿Todo el tiempo? No, sólo hasta que la mano coja la copa que hay sobre la mesa. En ese preciso instante, la copa que estaba oculta aparece y la que estaba visible en la mesa se oculta. Si hacemos coincidir la posición de ambas copas nadie se percatará del cambio. Por supuesto, a partir de ese instante la copa obedecerá a la mano en todos sus movimientos.

2. Medios de transporte: resulta muy frecuente que un personaje se introduzca en un medio de transporte (coche, autobús, moto, etc.) y se desplace a otro lugar. Esta situación resulta completamente opuesta a la que acabamos de explicar. Ahora es nuestro personaje el que es movido por un objeto aún mayor. ¿Cuál sería la forma más correcta de animar esta situación? Afortunadamente existen varias posibilidades, todas buenas, lo importante es conocerlas y utilizar la que mejor nos parezca.

La primera opción consiste en "clavar" (*Attach*) al personaje al objeto para que se mueva a la par que él. También podríamos agrupar ambos objetos de manera que se moviesen simultáneamente. Una tercera alternativa (aunque en este caso parece innecesaria) consistiría en segmentar la escena. Incluso podríamos utilizar *objetos guía* (*Dummy*) para aunar el movimiento de ambos objetos.

3. Seguir con la mirada a algo: imaginemos que una mosca se acerca a nuestro personaje y revolotea en torno suyo. Evidentemente el personaje la seguirá con la mirada. Pero, ¿cómo podemos hacer para conseguir que los ojos sigan el movimiento de la mosca sin salirse de las órbitas?. Esa es la cuestión.

Algunos paquetes incorporan una herramienta ideal para realizar este tipo de animaciones, pues permite constreñir la dirección de un objeto para que resulte apuntada hacia otro. Suele conocerse esa función con el nombre "mirar a". Ligando las pupilas del personaje a la mosca conseguiremos el efecto de que el personaje la sigue con su mirada, ya que dichas pupilas estarán siempre apuntando hacia la mosca.

Esta herramienta tiene diferentes nombres, dependiendo de cada paquete de software. Los más habituales son:

- En Softimage se encuentra bajo *Motion/Constraints/Direction*
- En 3D Studio MAX se denomina "*Look at*" y está en el *Track View*
- En ElectricImage se denomina "*Look at Object*" y está en el *Project Window*

DE NUEVO LA MEJOR JUGADA... PERÓ AHORA A SEIS BANDAS.



3D MAGIC MODELS

La libreria de modelos en 3D más ambiciosa del mercado.
Orientada para arquitectos, decoradores y constructores.
Su contenido formado por: Casa interiores, exteriores, Cocina, Living, Oficina, sillas, mesas, muebles.
Transportes, Mobiliario urbano. Modelos pre-texturizados.
Todos ellos en formatos: 3DS, DWG, DXF.

24.900
I.V.A no incluido



3D MAGIC TRANSPORTS

Sin duda la colección de transportes más potente hasta ahora realizada con una optimización excelente.
Divido en las secciones de Tierra, Mar, Aire
Tecnologia Ready-to-Use, los detalles al igual que el objeto son totalmente en 3d.
Formatos: 3DS, MAX, LWO, DWG,DXF.





3D MAGIC MUEBLES&DECORACION

Alguna vez habia visto un mueble perfectamente modelado en 3D!. Pues ahora disponible una colección de muebles de hogar, cocina y baño, para que no tenga que construirlos. Gran variedad y diversos estilos.
Modelos Pre-texturizados con maderas auténcticas de cada mueble.
Formatos: 3DS, MAX, DWG, DXF

28.400
I.V.A no incluido



PHOTOSYNTHESIS

Le ofrece una libreria de objetos 3D basada en modelos naturales. Su contenido, una colección de arboles y plantas de una gran calidad, y facil manejo, gracias a su optimización de poligonos.
En ella tambien podrá encontrar un gran surtido de composiciones de maceteros de distintos tipos, enredaderas etc... que solo tendra que colocar en su escena donde guste, llegando a llenar sus renders de vida natural.
Formatos: 3DS, MAX, DWG, DXF.

14.500
I.V.A no incluido



BITMAP LIBRARIES

Bitmap Libraries The Series y su paquete le ofrece miles de texturas e imágenes 2D para utilizarlas como material de soporte en sus programas de arquitectura, decoracion, video-producción etc...
En su interior centenares de marmoles, revestimientos, pavimentos, metales, maderas, zocalos, transportes, mosaicos, tejados, plantas, cielos...
Formatos soportados: TGA.

17.400
I.V.A no incluido



BITMAP TEXTILES

Bitmap Textiles le ofrece una colección de 250 texturas textiles retocadas minuciosamente, para ser usadas como revestimientos de sofas, cortinas y todo tipo de tapizados.
Su contenido incluye: rugosos, lisos, acuadros, estampados, etc.
Formato: TGA, TIF.

7.500
I.V.A no incluido



c/Caballero 79 5ª Planta.
08014.BARCELONA-SPAIN.
Telf/Fax: (93) 439 53 02
Intl: +34 3 439 53 02
E-MAIL: 4BYTES@4BYTES.COM
HTTP://WWW.4BYTES.COM

3D DATA SHOP
La tienda de objetos 3D en internet en la cual podrá adquirir modelos de forma unitaria.





3D STUDIO

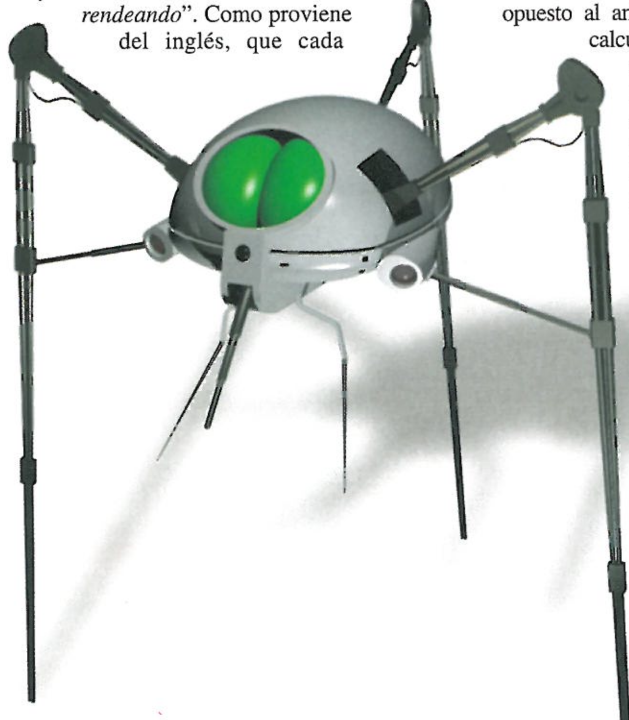
Opciones de Render
Autor: Guillermo Gómez

Nivel: Medio

El proceso de render genera la imagen o animación final. Su duración suele ser directamente proporcional a la complejidad de la escena. La representación final que se obtiene es configurable por el usuario mediante una serie de parámetros. El manejo correcto de éstos debería tener como consecuencia una calidad adecuada a las necesidades, pero ponderada por el uso de los recursos disponibles.

Normalmente, el deseo de ver realizado un trabajo de muchas horas tiene más importancia que la sensatez. Se tarda entre uno y diez minutos, como máximo, en configurar correctamente un render y, de no hacerlo, perdemos, además del tiempo del proceso, entre quince y veinte minutos para descubrir qué ha pasado, corregirlo y volver a repetir el proceso. Después de leer las configuraciones y posibilidades del render, todo ese tiempo que deberíais ganar tendrías que gastarlo en algo provechoso, como salir a la calle, pasear y observar el paisaje sin pensar en qué bien quedaría en modo *Wire*.

El término render, por desgracia, es casi intraducible al castellano; lo más aproximado sería llamarle representación, pero prácticamente nadie lo usa, y en este mundillo de las 3D se puede oír desde un "*renderízame esta frame*", hasta "*me tiré tres horas rendeando*". Como proviene del inglés, que cada cual lo use bajo su responsabilidad, pues todo el mundo lo entenderá más o menos, excepto los profanos, que se les reconoce rápido por el gesto de arrugar la nariz. Una vez hechas las salvedades lingüísticas pertinentes podemos entrar en materia.



El proceso de render consigue transformar nuestras mallas, luces y materiales en algo coherente para el gran público. Hay muchos programas de render; cada uno lo calcula de una manera distinta, aunque en el fondo se aplican a todos unos mismos principios físicos y matemáticos. Entre las variedades más comunes esta el *Raytracing* o trazado de rayos de luz, que comprueba la trayectoria de los rayos, qué objetos se interponen entre ellos, cuáles reflejan la luz y cuáles la absorben. En fin, un proceso muy complicado de cálculo. En 3D Studio 4 se usa el método de *Radiosity* que es, exactamente, el opuesto al anterior, es decir, se calculan las trayectorias desde el objeto al observador; en este caso, sería la cámara o la vista desde la que estuviéramos situados.

En el menú *Renderer* tenemos, inicialmente, la posibilidad de renderizar una vista completa mediante *Render/View*, o bien una zona o región con *Render /Region*, que definiremos marcando las dos esquinas que forman el rectángulo que enmarcará la zona. Con esta última opción, el render será del tamaño proporcional al área que hayamos definido, pero si deseamos que esa zona marcada, una vez renderizada, ocupe toda la pantalla deberemos usar la opción *Render Blow up*. También podemos visualizar, si así lo deseamos, un solo objeto con *Render Object*, que permite ver con detalle todas la propiedades que tenga sin necesidad de visualizar los otros. Esta opción se usa mucho cuando a una escena definitiva se le añade algún objeto. Primero, usando este comando nos aseguramos que el objeto se comporta adecuadamente, y, después, lanzamos un render global de toda la escena para asegurarnos de que las luces las sombras, etc, están todas en su sitio y no se produce ningún comportamiento anormal.

Con el comando *Render Last* realizamos una representación del último lanzado, sea del tipo que sea; es práctica para descubrir el efecto de modificaciones de última hora sin necesidad de variar la zona, o los parámetros que hubiéramos establecido.

CONFIGURACIÓN

Una vez vistos los diferente tipos de representaciones que podemos obtener debemos pasar al cuadro de *Setup*, donde vamos a establecer las peculiaridades de nuestro render.

La primera opción *Renderer/Setup/ Atmosphere* se encarga, como su propio nombre indica, de configurar y definir todos los efectos de tipo atmosférico que va a poseer la escena. Para definir correctamente estos efectos y su alcance se necesitan muchas pruebas; no es fácil acertar exactamente con el efecto que teníamos en mente, pero intentaremos señalar unas aproximaciones para acercar las posibilidades de cada parámetro. El primero, *Fog*, se encarga de dar el aspecto de niebla aunque, puntualizando más, decir que produce un efecto más cercano a lo que se conoce como neblina, es decir, conforme aumenta la distancia, los objetos se van desvaneciendo para confundirse con el color del fondo. Tras activar el botón, podemos pasar a configurar-

OPCIONES DE FONDO

Hemos hablado mucho del fondo o background; éste se establece mediante el menú *Renderer/Setup/Background*. Tras pulsarlo, aparecerá una caja titulada *Background Method*, mediante la cual podemos seleccionar un color único con *Solid Color*, o un degradado de hasta tres colores con *Gradient*. Aquí, no sólo podemos definir los tres tonos que aparecerán en el gradiente, sino que incluso podemos ampliar la posición del central pinchando en el cuadro donde se representa el degradado. Un tipo de background muy interesante es el Bitmap, con el pegamos en el fondo una imagen de un archivo de mapa de bits. Ya vimos que esto es útil, no sólo de cara al render sino que, además, podemos visualizar esa imagen para hacer un dibujo a mano alzada sobre ella, o para ajustar la perspectiva con el *Camera Control*. Es importante recordar cuando ponemos un Bitmap como fondo, que en el cuadro de diálogo del proceso de render podemos reescalar este Bitmap para que ocupe totalmente el fondo, o mostrarlo como mosaico con la opción *Tile*. Ante la posibilidad de reescalarlo, es muy importante tener en cuenta que si la imagen se amplía demasiado se producirá un desagradable efecto de pixelado que tirará por tierra nuestra escena. A veces, por tanto, es preferible intentar dimensionar de nuevo la imagen mediante un programa de retoque fotográfico que, con ayuda de sus filtros, conseguirá reducir la pérdida de calidad que se produce cada vez que ampliamos una imagen.

la pinchando en *Define*, donde nos aparece un cuadro de diálogo con varias posibilidades. Nos encontramos con las seis barras para definir tonos; las tres primeras, R, G, B corresponden a las iniciales de *Red, Green, Blue*, los tres colores primarios, rojo, verde y azul que, todos a nivel máximo, dan blanco, mientras que al mínimo producen negro. Las tres siguientes H, L, S, corresponden a *Hue, Luminance* y *Saturation*, es decir, tinta, iluminación y saturación, o porcentaje de tinta. Bien, pues con estas seis barras conformamos el tono de nuestra niebla. Debajo, con los porcentajes de *Near*, cercano, y *Far*, lejano, determinamos la cantidad de color de niebla que recibirán los objetos que están en el plano más lejano o más cercano con respecto a la cámara. Recordemos que estos dos planos los establecemos gráficamente mediante el comando *Camera/Ranges*. Si, además, activamos el botón *Fog Back ground*, la imagen que hubiéramos situado como background, o fondo de la escena, también recibirá el tratamiento de fusión con el color de la niebla en la proporción que hayamos establecido en *Far*. Con ello conseguiremos un mayor realismo en la escena en el caso de que hayamos incluido la foto de un paisaje como background, por ejemplo.

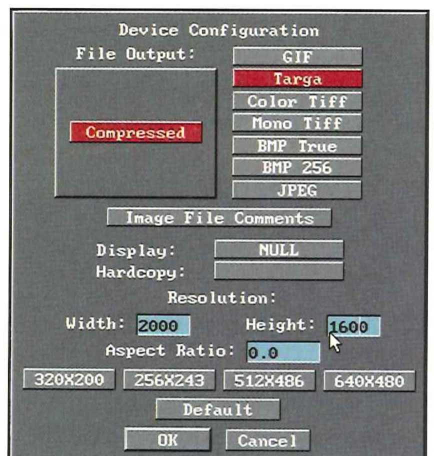
El siguiente efecto atmosférico, *Distance Cue*, funciona de manera muy parecida a la niebla pero con la salvedad de difuminar los objetos hacia un solo color, el negro. Tras pinchar el botón *Define* aparecen nuevamente los mismos parámetros de porcentaje *Near* y *Far*, que actúan de idéntica manera. El botón *Dim Background* también es similar al caso anterior sólo que, esta vez, aplicará el porcentaje de *Far* a la imagen de fondo, oscureciendo la imagen, difuminándola hacia el negro. Evidentemente, esta opción tiene aplicaciones muy prácticas cuando manipulemos escenas con escasez de luz queriendo crear ambientes lóbregos y tenebrosos.

El último efecto atmosférico de que disponemos se llama *Layered Fog*, algo así como niebla por capas. En realidad se trata de una niebla de desarrollo vertical, es decir, de arriba abajo o viceversa, donde vamos a usar las coordenadas reales de la escena. Para ello, definimos mediante el parámetro *Top* el límite superior exacto, en coordenadas de la niebla; mediante *Bottom*, elegimos el límite inferior, y con *Density* especificaremos la cantidad de niebla, en donde cero indica que no habrá niebla y 100 que ésta será opaca. Al lado, tenemos la característica *Falloff* que va a establecer la dirección en la que la niebla se espesa. Con *Top* la niebla comienza en la parte inferior transparente para alcanzar, en la zona superior, la máxima opacidad que hayamos definido en *Density*. Si resaltamos la opción *Bottom* sucederá al revés, esto es, alcanzará su máximo espesor en la coordenada definida como inferior. *None* actúa dejando toda la niebla con la misma densidad. Por su parte, el botón *Fog background* actúa como en el resto de las operaciones atmosféricas.

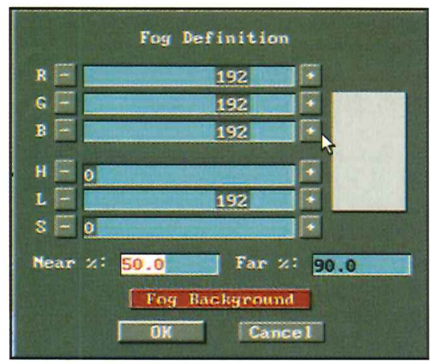
Para configurar el tipo de salida que tendrá el render tenemos el menú *Renderer/Setup/Configure*; tras escogerlo, aparece una ventana titulada *Device Configuration*. En ella podemos encontrar todos los tipos de archivos gráficos de salida que nos permite 3D Studio, que son: GIF, Targa con extensión .TGA, TIFF en color o monocromo, esto es blanco y negro, BMP True o real de 24 bits, BMP de 256 colores y JPEG. Cuando el fichero gráfico admite compresión nos lo hace saber en el cuadrado de la derecha, donde pinchando nos permite decidir si lo comprimimos o no. En el caso particular de los JPEG, podemos variar la relación de compresión que deseamos aplicar pero, como muy gráficamente nos muestra el programa, a mayor compresión, mayor pérdida de calidad. Otra posibilidad que tenemos, aunque no está recogida aquí, es la de generar una salida Postcript, es decir, en formato .EPS. Este formato es muy útil para determinadas impresoras que poseen este lenguaje, pues facilita mucho la impresión y, sobre todo, la calidad. Dejando a un lado éste, desde estas páginas recomendamos el uso del formato Targa a aquellos que no tengan complejos a la hora de ocupar disco duro, y el JPEG, al resto de los mortales que, sin llegar a una excesiva pérdida de calidad, consigue unos tamaños muy mínimos para lo que se estila en los archivos de gráficos. Éste es, sin duda, el motivo por el que este formato se está imponiendo cada día más para la transferencia de gráficos en la gran malla.

Siguiendo en la ventana de configuración tenemos la posibilidad de añadir algún que otro comentario a la imagen de salida pulsando el botón *Image File Comments*. Esto sólo lo admiten los ficheros de 24 bits, como el Targa y el BMP True. Después de escribir el nombre del fichero en el selector, nos aparece el recuadro para introducir el texto. En los dos siguientes apartados, *Display* y *Hardcopy*, podemos escoger el tipo de Driver que va a usar el programa para representar la imagen en pantalla o a través de una impresora. Anteriormente a esto, habremos tenido que configurar estos drivers e instalarlos la primera vez que ejecutamos 3D Studio mediante el arranque de VIBCFG.

CONFIGURACIÓN DE SALIDA.



CAJA DE OPCIONES.



EFECTOS ATMOSFÉRICOS.

El último apartado es la resolución de la imagen. Tenemos cuatro botones con resoluciones preestablecidas, o la posibilidad de introducirla nosotros a mano con los cuadrados *Width,* o ancho y *Height* o alto. En cuanto a Aspect Ratio define la proporción de pixels en la representación en pantalla, si se deja a cero, el programa lo calcula automáticamente. En este apartado señalar algo importante, si deseamos crear una imagen con una resolución mayor que la que los drivers de nuestra pantalla permiten, pongamos por ejemplo 2000x1600, debemos desactivar la pantalla, para que no se bloquee al intentar representarla; esto lo hacemos con la opción *NULL* en los drivers de *Display*. Por supuesto, no debemos olvidarnos de activar la salida a disco del render o nos encontraremos con el cálculo de una representación que no va a ser guardada y tampoco visualizada, trabajo para nada en otras palabras. A veces también se activa la opción *NULL* para ahorrar tiempo de render, si bien esto lo comentaremos un poco más adelante.

OPCIONES DE RENDER

El siguiente menú de configuración es *Renderer/Setup/Options* que, al escogerlo, aparece una ventana titulada *Render Options* con las siguientes posibilidades:

• *Object Motion Blur*: Sería algo así como borrosidad de objeto en movimiento. Para los entendidos en fotografía se puede resumir como el efecto obtenido cuando la velocidad de obturación es incapaz de fijar un sujeto moviéndose. Para los no entendidos crea una imagen ligeramente borrosa de los objetos en movimiento, consiguiendo así dar una mayor sensación de velocidad. No actúa sobre todos los objetos, sino sobre aquellos que hayamos definido en el módulo Keyframer con esta característica mediante el comando *Objects/ Motion Blur*. Un ejemplo muy claro del uso de este comando sería, por ejemplo, para aplicar a las aspas de un helicóptero, que nunca llegan a verse claramente cuando están en movimiento sino que generan una especie de círculo borroso al rotar. Para personalizar este efecto disponemos de tres variables. La primera de ellas es *Number*, que podemos establecerla entre dos y ocho, y especifica el número de imágenes que se superponen en una sola. Es decir, si dejamos el valor por defecto de siete esto conseguirá que, en la primera, secuencia o frame de la animación aparezca el objeto, y superpuestas, sus siete siguientes posiciones en las próximas frames. Mediante *Samples* establecemos el espaciado entre las copias del objeto y, obviamente, no puede alcanzar un valor superior al de *Number*. Con *Duration* definimos el tiempo en cuadros durante el que se va a mantener abierta la cámara.

• *Scene Motion Blur*: Produce el mismo efecto de borrosidad pero afectando a toda la escena con la proporción que indiquemos en *Dither*.

• *Render Fields*: Se usa sólo para volcados directos a vídeo profesional; se conoce como renderizar por campos, y produce dos renders por fotograma que consiguen una mayor calidad de movimiento. Actualmente, está un poco en desuso, pues podemos editar frame a frame en programas específicos de edición de ví,deo, desde los que después se pasa mediante la tarjeta hardware adecuada a cinta. No se debe usar nunca para generar archivos de animación.

• *Video Color Check*: Es un método de corrección de color para adecuar los que genera el ordenador a los estándares PAL o NTSC; corregirá hacia uno u otro dependiendo del que tengamos puesto en el fichero 3DS.SET. Tenemos dos métodos de corrección: *Flag*, que sustituye los colores no válidos por el negro, y *Correct*, que los aproxima lo más posible entre sí. En principio, al igual que en el apartado anterior, la opción *Video Color Check* debería estar desactivada siempre que no estemos haciendo un volcado directo a vídeo dado que, mediante un programa editor, deberíamos poder hacer estas correcciones si fuese necesario.

• *Dither True Color*: Con esta opción se establece la conversión a colores reales. Debe estar activada casi siempre.

• *Super black*: Para el fondo de la escena usa un color negro muy intenso, mientras aleja los colores más oscuros de la escena del negro. Esto se usa para componer vídeo luego sobre el fondo, salvo para esta opción, por lo que no es en absoluto recomendable.

• *Dither 256*: Es muy recomendable tenerla activada cuando estemos generando imágenes de 256 colores, pues evita ese espantoso efecto que consiste en que los degradados se ven formados por bandas y sin uniformidad.

• *Reflect Flip*: Activada, invierte los mapas de reflexión.

• *Render Alpha*: Permite generar un canal de transparencia de la imagen, lo que se conoce como canal Alpha, muy útil para componer el resultado del render con otras imágenes. Añade 8 bits más a las imágenes de tipo Targa donde va la información sobre la transparencia.

• *Alpha Split: Separar Alpha*: Se usa siempre y cuando hayamos activado la opción Render Alpha pues, con ésta, conseguimos obtener dos ficheros distintos: por un lado, el usual, y, por otro, uno adicional con la información de transparencia del anterior. Se graba con el mismo nombre pero anteponiendo un A_.

• *TGA Depth*: Profundidad del formato TGA que será de 16 o 24 bits; con 16 bits no nos permitirá crear un canal Alpha, pero también ocupará menos espacio que el de 24. Debemos establecerlo dependiendo del uso que vayamos a dar a la imagen generada.

• *Save Last Image*: Guarda la última imagen generada.

• *Nth Serial Numbering*: Esta opción se usa cuando, en vez de generar como salida un fichero de animación, se graba cada fotograma de forma independiente. El programa asigna un nombre a cada fichero, con un número que corresponde a la frame equivalente. Si hemos decidido que los fotogramas a renderizar no sean correlativos, sino que vayan produciéndose saltos con el parámetro *Frame Step* generando, por ejemplo, los pares, con esta opción activada les añadirá los números a los ficheros sin contar el salto, esto es, en orden y aumentando uno en cada fotograma.

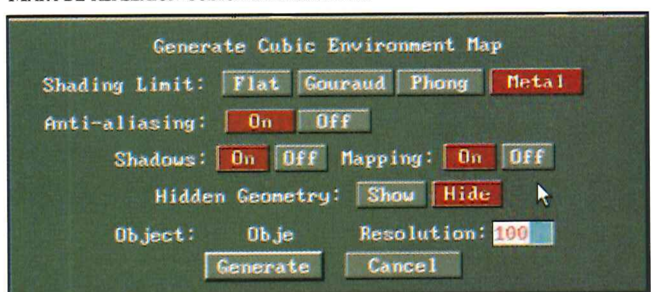
• *Z Clip Near*: Marca la distancia en unidades de coordenadas desde la cámara al primer plano que se va a visualizar. Todo lo que haya entre medias no será representado.

• *Safe Frame*: Si en el menú *Views* activamos la opción *Safe Frame* nos aparecen en la vista activa dos rectángulos, uno de color amarillo, que muestra la parte de la imagen que se va a visualizar en pantalla, y otro verde, interior, que mostraría la parte que definimos con este comando. Esto es práctico para delimitar zonas en las que se va a desarrollar la animación, o para atenernos a unos márgenes como, por ejemplo, el que ocuparía la animación en una pantalla de televisión.

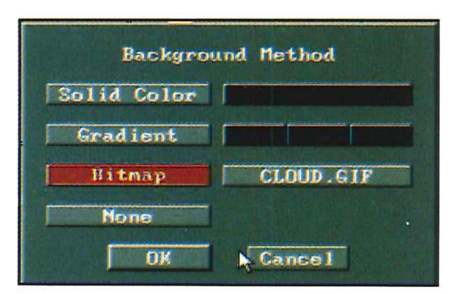
• Pixel Size: Es el tamaño del píxel. Varía entre 1 y 1.5; cuanto más grande fuese mayor calidad se debería obtener, pero también mayor tiempo de render.

Ya tenemos todas las opciones configuradas; bastaría con pulsar *OK* para activar los cambios o *Cancel* para dejar las opciones que traía por defecto. En general, y salvo casos

MAPA DE REFLEXIÓN CÚBICA DEL ENTORNO.



CONFIGURACIÓN DE SOMBRAS.



ESTABLECIMIENTO DE FONDOS.

especiales, estas opciones no se suelen alterar y los valores por defecto son aplicables para la mayoría de animaciones, pero siempre es bueno conocerlas, saber que están ahí y que podemos usarlas en cualquier momento.

LAS SOMBRAS

La siguiente opción a configurar es el control de sombras; se usa el menú *Renderer/Setup/Shadows*. Tenemos dos formas de generar sombras: mediante mapas o mediante trazado de rayos. Para generarlas mediante mapas activaremos la opción *Shadow Maps*, que dispone de tres variables a configurar. Con la primera de ellas, *Map Bias*, establecemos el desplazamiento de la sombra con respecto al objeto. Con *Map Size* determinamos el tamaño del mapa de sombra, en píxeles. Si el tamaño es muy grande, el mapa tendrá mayor resolución pero influirá notablemente en el tiempo de render. La última variable, *Map Sample Range*, establece la dureza o suavidad de las sombras; dicho de otra forma, establece la cantidad de zona de penumbra que tendrá el borde de la sombra. El valor puede variar entre 1 y 10. La generación de sombras más real se produce cuando se selecciona la opción *Ray-Traced Shadows*, pero por la cantidad de tiempo que necesita durante el tiempo de render es muy necesario estar completamente seguro de la necesidad de uso de esta opción. La única variable a definir es *Ray Trace Bias* que coloca la sombra delante o detrás del objeto.

El último menú del *Setup* genera los mapas de reflexión cúbica del entorno. Tras pinchar la opción *Renderer/Setup/ Make .CUB* el programa nos pide el objeto que se usará como centro de la proyección. Una vez elegido el objeto aparece el cuadro de diálogo titulado *Generate Cubic Environment Map*, en el que establecemos mediante *Shading Limit*, el tipo de render que va a generar si la imagen se va a calcular mediante los sistemas *Flat, Gouraud, Phong,* o *Metal*. También podemos decidir si deseamos que se aplique un efecto Anti-Aliasing a las imágenes, si bien esto sólo es recomendable cuando las imágenes generadas sean de un gran tamaño, y necesitemos mucha calidad. Es importante recordar que estos archivos se van a usar como mapa de reflexión, es decir, salvo excepciones, únicamente son para generar la sensación de que un objeto refleja su entorno, así que, por norma general, no necesitaremos imágenes de una calidad y nitidez espectacular. Esto es aplicable a las dos siguientes opciones: *Shadows*, que determina si en el render se van a calcular las sombras, y *Mapping*, si se va a usar los mapas para ajuste de texturas. Todo dependerá del uso que vayamos a dar a nuestro mapa de reflexión cúbica de entorno. La siguiente opción, *Hidden Geometry,* nos pregunta acerca de la posibilidad de mostrar la zona oculta de los objetos. Por último, la ventana nos informa del objeto que hemos elegido como centro de la proyección, y nos permite establecer la resolución en píxeles del mapa. Esta resolución vendrá dada, obviamente, por el tamaño del objeto al que se le van a pegar estos mapas; si el objeto es pequeño, ponerle una dimensión tremenda sólo va a conseguir un render mastodóntico.

Una vez configurado el cuadro, pulsando la tecla *Generate*, deberemos introducir un nombre, con un máximo de seis caracteres, para que el programa calcule las seis imágenes, añadiéndole a cada una dos letras que determinarán la posición de la imagen con respecto al archivo; estas serán: arriba, up, abajo, dn, izquierda, lf, derecha, rt, delante, ft, y detrás, bk, que corresponden a las siglas corresponden a UP, DOWN, LEFT, RIGHT, FRONT, y BACK. Además se genera también un archivo .CUB que será el que asignemos en el *Materials Editor* como mapa de reflexión. Recordemos que todos estos archivos deben estar en las rutas definidas para los ficheros de mapas. El fichero .CUB está en formato ASCII y es, por tanto, editable y modificable para experimentar un poco con él.

Ya conocemos prácticamente todas las posibilidades de render, pero llegado el momento de lanzarlo, aún aparecen más posibilidades de modificación para ajustarlo como un guante, que es de lo que se trata.

Algunas de estas posibilidades son:

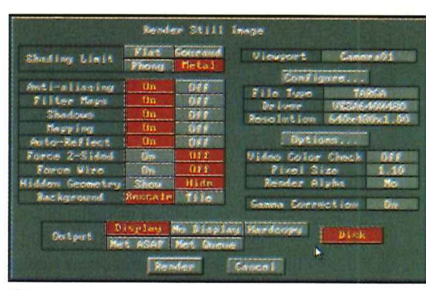
• *Shading Limit*: Es el tipo de sombreado con que se va generar la imagen. Hay cuatro tipos *Flat, Gouraud, Phong* y *Metal*. Los dos de más calidad son *Phong* y *Metal,* aunque por supuesto son los que más recursos necesitan

• *Anti-Aliasing*: Podemos activarlo para evitar el desagradable efecto de escalonamiento que se produce a veces entre píxeles cercanos con una gran diferencia de tonalidades.

• *Filter Maps*: Debemos activarla siempre que vayamos a realizar un render definitivo. Consiste en un suavizado mediante filtros de los mapas de textura aplicados a los objetos. Sólo debemos desactivarla cuando generemos pruebas con objeto de acelerar un poco más el proceso.

• *Shadows*: También podemos activar y desactivar las sombras a nuestro antojo. Por supuesto, cuando estén desactivadas no se toma en consideración nada del apartado en que hemos definido el tipo de cálculo de sombras.

• *Mapping:* Activamos los mapas de situación de las texturas, no afecta a los mapas de reflexión pues estos no necesitan mapeado de ningún tipo.



VENTANA DE RENDER.

• *Auto-Reflect*: Desactivando los mapas de reflexión automática ganaremos una gran cantidad de tiempo en pruebas. Es interesante descubrir que si abusamos de estos mapas el render puede hacerse, como mínimo, interminable.

• *Force 2-Sided*. Esta opción sólo debemos activarla cuando tengamos objetos con texturas de este tipo, en la que se nos muestran las dos caras del objeto; esta característica la llevan sobre todo los materiales de tipo transparencias.

• *Force Wire*: Si se activa la escena será renderizada en modo malla o *Wire*, ignorando la mayoría de los materiales de los objetos.

• *Hidden Geometry*: Con la opción *Hide* activada el render sólo afectará a aquellos objetos que no estén ocultos, mientras que con *Show* el render se efectuará de absolutamente toda la escena.

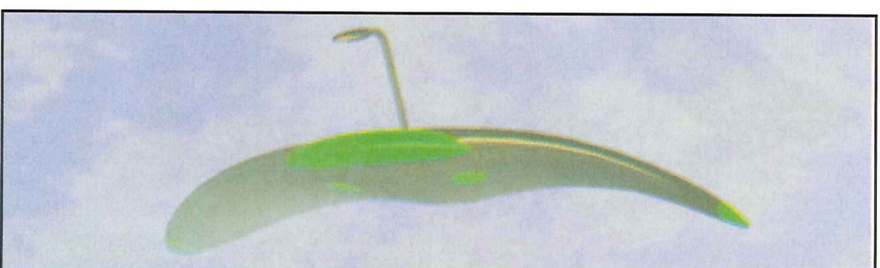
• *Background:* Aquí definimos si deseamos que la imagen que hayamos colocado como fondo se reescale o se utilice como mosaico, con *Tile*, para cubrir todo el fondo. Es necesario pensar que si la imagen a reescalar es pequeña con respecto a la resolución final de salida, el render necesitará más recursos para generar la imagen.

• *Configure y Options*: Nos llevan a las mismas ventanas que *Renderer/Setup/ Configure* y *Renderer/Setup/Options*. Por si se nos había olvidado definirlas o preferimos configurarlas a pie de pista.

• *Output*: Establece la salida del render. Puede ser por pantalla con *Display* activado, por impresora, con *Hardcopy* activado, a disco, con *Disk*, o a una red en caso de que estemos en ella con *Net ASAP,* y *Net Queue*. La más interesante es *No Display,* que equivale a utilizar el driver *Null* de salida en la opción *Configure*, se ahorra algo de tiempo de render y, además, es útil cuando la resolución de la imagen generada supera a la que tenemos definida en pantalla. Es muy importante no olvidar nunca activar la salida a disco, para poder ver la imagen.

Ya tenemos el render configurado ahora sólo queda observar durante horas la interminable evolución de la barrita roja, nuestra peor pesadilla.

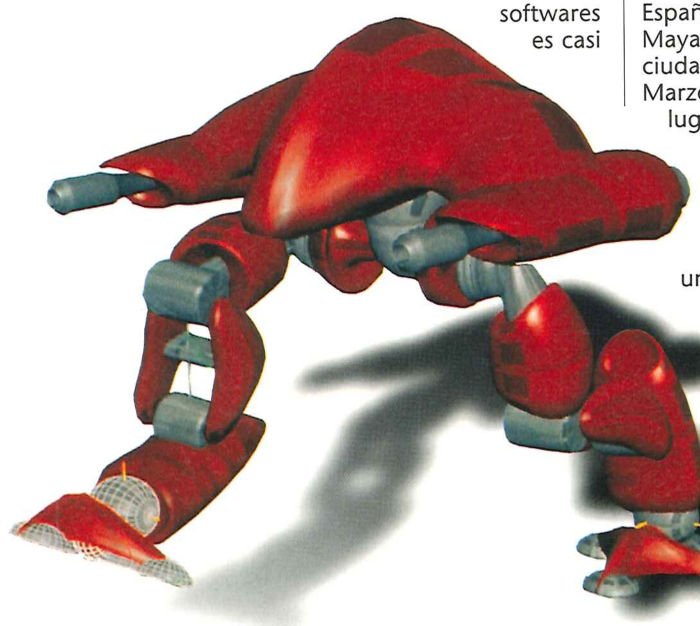
EJEMPLO DE EFECTO NIEBLA.



Maya, una nueva

Después de una larga espera, el nuevo programa de Alias|Wavefront es una realidad. Hace unos pocos meses, Maya hizo acto de aparición en el Broadcast, donde se pudieron ir apreciando las líneas maestras sobre las que se iba a asentar. SGO, Soluciones Gráficas por Ordenador, representante oficial en exclusiva para España de Alias|Wavefront, ha sido la encargada de la presentación en nuestro país.

Tres años de desarrollo han dado como resultado un programa nuevo. Lo de nuevo se ha de entender como algo más que una de las múltiples revisiones a las que se sometía a Power Animator. Maya sigue teniendo una filosofía de trabajo similar a Power Animator, pero nada más. Basta con comparar ambos interfaces para darse cuenta de ello. De todos modos, aunque la transición entre estos dos softwares es casi obligada, esta pasa por una convivencia temporal.



Maya World Tour

El largo periplo que está haciendo Maya por toda Europa y por algunos países de Africa y Oriente viene a confirmar el interés de la compañía en dar una completa información a todos los profesionales que llevan trabajando con ellos y, por otro lado, el de extender lo máximo posible su oferta a todo aquel que esté metido de lleno en este mundo. En España, la presentación de Maya se ha hecho en dos ciudades. El miércoles 4 de Marzo, la presentación tuvo lugar en Barcelona para, un día mas tarde, hacerlo en Madrid. La cita tuvo lugar en el Palacio de Fernán Nuñez, y congregó a un importante número de personas relacionadas con el sector. Después de recoger la acreditación, nos fueron acomodando en una sala donde se había instalado una gran pantalla. Mientras comenzaba la demostración y se proyectaban imágenes correspondientes a películas, anuncios, etc., llenas de efectos especiales, fuimos echando un ojo al dossier que se nos entregó a cada uno.

El dossier constaba de varias partes, un cuadernillo con características técnicas y con apuntes de la demostración que más adelante se iba a llevar a cabo, otro, en castellano, con información algo más general y que venía a complementar al anterior, y un tercero, preparado por la importadora, con información general y con un apartado que reflejaba el desarrollo comercial de la marca en España. Aunque a primera vista no le presté mucha atención, al revisar mis notas para escribir este artículo, me pareció necesario destacar algunos datos que reflejan el grado de penetración de estos productos en nuestro país.

El período del que se está hablando es de unos cuatro años. Pues bien, en estos cuatro años el incremento de la facturación ha sido espectacular, tanto en productos de Alias como en programas tales como Jaleo, Tic-Tac Toon o 3D Equalizer. Siempre según los datos de SGO, de unas insignificantes y anecdóticas veintitres mil seiscientas pesetas en el primer trimestre, y de algo mas de cincuenta y ocho millones a

generación

finales de 1994, se ha pasado a los setecientos treinta y tres millones de finales del noventa y siete. Esto nos lleva a deducir que la estructura empresarial que se ha creado en este período constituye un asentamiento sólido de Alias|Wavefront y, por ende, de Silicon Graphics.

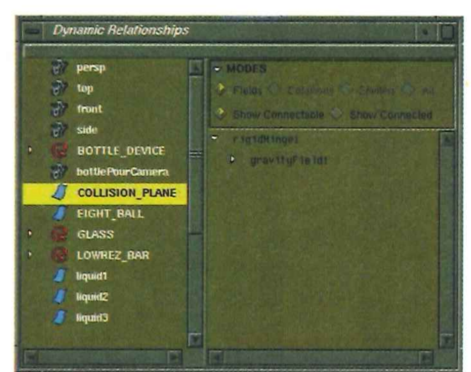
Como todos sabemos bien, uno de los inconvenientes a los que se ha tenido que enfrentar la industria profesional del 3D han sido unos costes ciertamente elevados a los que sólo podían tener acceso sociedades con un fuerte respaldo capital. Esto dejaba un poco fuera de mercado a iniciativas con presupuestos más reducidos, las cuales optaban por otros equipos más acordes con sus posibilidades económicas. Parece ser que la política de precios que se lleva siguiendo desde hace una temporada está dirigida a combatir este aspecto y que Maya, por lo que se allí se dijo, va a seguir en la misma línea de actuación.

Uno de los conferenciantes fue Colin Brown, manager general de la prestigiosa compañía británica Cinesite. Él aportó el punto de vista empresarial desde la perspectiva de una compañía que empezó con una plantilla de tres personas y que actualmente se compone de unas cien. Como es fácil de imaginar, la base informática sobre la que asienta esta empresa está en manos de productos de Alias, aunque Colin dijo estar hablando desde una perspectiva personal alejada de posiciones corporativistas. La razón por la que argumentó confiar en Alias es por la constante renovación a la que son sometidos todos sus productos, con lo que consiguen permanecer en los puestos de cabeza de toda la industria de software infográfico.

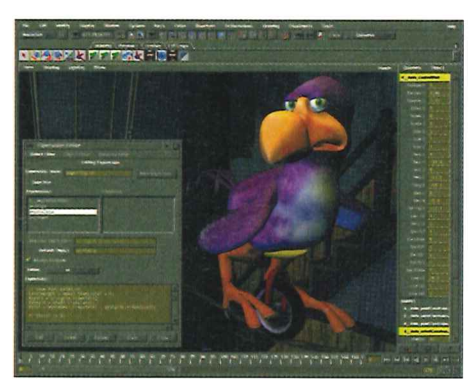
Rondando las diez y media, el director comercial de la oficina de Barcelona, Xavier Pampols, inició la presentación. En primer lugar, presentó a todos aquellos que iban a intervenir en el acto y agradeció a los componentes de SGO Madrid la organización del evento. Comenzó con la explicación del organigrama de la compañía, de sus objetivos y de la historia de ésta a la que se hacía referencia unas líneas mas arriba. A continuación, presentó a Peter Gambirasio, delegado de Alias|Wavefront para España, el cual fue dando ya algunos datos específicos sobre Maya. Habló de la distribución del programa, de sus equivalencias con Power Animator y, ya en la parte final, contestó a algunas preguntas que le hicieron alguno de los asistentes. Por cierto, una de las preguntas que, según él, se le planteaba una y otra vez, era sobre el lanzamiento de una versión para NT. En la respuesta vino a decir que, aún siendo uno de los proyectos sobre los que se está trabajando, el futuro de este aspecto está algo lejos todavía, dejando caer que, lo que si era seguro, iba a ser la necesidad de equipos que aseguraran la operatividad de un programa tan exigente como Maya.

En cuanto a los plazos de distribución, no quiso dar ni fechas ni ningún dato concreto amparándose en la dificultad que plantea distribuir cerca de dos mil licencias para toda Europa, aunque confiaba que esto se solucionara en un plazo razonable. Otro aspecto que interesaba mucho era las equivalencias existentes entre Maya y Power Animator. En esto y en los precios sí se concretó la información. Los precios aparecieron en una lista reducida con las opciones y configuraciones más comunes, lo cual sirvió para hacerse una idea de por dónde iban a ir los tiros.

Otra de las cosas a las que Peter concedió una importancia especial fue a la nueva arquitectura de Maya y a las ventajas que ofrece. Las prestaciones gráficas han crecido y han mejorado ostensible-



Ventana que muestra las relaciones dinámicas entre objetos.


El Editor de Expresiones de Maya.

Relación entre Maya y Power Animator

Power Animator	Maya
Power Animator Base	Maya Power Modeler
Power Animator Modelado Avanzado	Maya F/X
Power Animator Animación Avanzada	Maya Artisan




Imagen de una luz emitiendo partículas.



HyperGraph y *Action Window*.

Desde el comienzo todo parece nuevo, el interfaz hace que la pantalla quede bastante liberada y se aproveche en su totalidad. Las opciones están agrupadas en un gran menú que funciona como los antiguos *Marking Menus* de *Power Animator.*

mente gracias a la estructura OpenGL. En este mismo apartado, MEL (*Maya Embedded Language*), es otra de las bases sobre las que se fundamentan algunas otras mejoras, ya que MEL es un lenguaje de programación abierto con el que se puede acceder a todas las prestaciones internas de Maya. Se insistió en la rentabilidad que proporciona el trabajo en común del animador y del director técnico a la hora de determinar funciones específicas en proyectos donde éstas sean necesarias, o también a la hora de optimizar tareas rutinarias, etc.

La posibilidad de crear Plug-ins para Maya está también definida por el entorno MEL. Todo el desarrollo del programa se ha hecho con este lenguaje, y de ahí esta capacidad que, unida a la disponibilidad de un API C++ llamado OpenMaya, ponen al alcance de cualquiera la mejora en la productividad con un esfuerzo mínimo.



Aspecto de una mano realizada en Maya con gran realismo.

Peter Gambirasio se despidió de los asistentes bromeando sobre la cara que se nos iba a quedar cuando presentaran la que sin duda iba a ser la estrella de la mañana, Artisan, y nos emplazó al turno de preguntas que iba a haber una vez finalizada la charla técnica que iba a dar Javier Moreno.

Tras unos minutos de receso, se convocó de nuevo a todos en el interior de la sala para continuar con la segunda parte del acto. Javier Moreno es el director técnico de SGO y, al igual que en el Broadcast, se encargó de ir explicando los pasos de la demostración que él mismo realizaba sobre una plataforma Octane. Primeramente , se proyectaron dos cortos de animación. Uno, el de una conversación entre un payaso, Bingo, y un joven que contestaba a las preguntas del payaso en una sucesión de escenas de lo mas surrealista. Al igual que en las anteriores animaciones creadas por el equipo de Alias, las historias siguen siendo poco convencionales, pero sus personajes humanos avanzan meteóricamente en calidad y en el grado de realismo. En la historia de Bingo y el tipo de la gorra, ambos tienen un nivel de acabado realmente bueno, soportando primeros planos sin ningún problema. Sin embargo, la segunda de ellas, que es la que protagoniza un cowboy llamado Mel y que fue la que se vio en el Broadcast, se rompe con la tradición y es básicamente una historia de dibujos animados realizada con modelos 3D.

Con ayuda de Mel y del pájaro Dodo se fueron analizando un montón de nuevas soluciones y nuevos conceptos de Maya.

Desde el comienzo todo parece nuevo, el interfaz hace que la pantalla quede bastante liberada y se aproveche en su totalidad. Las opciones están agrupadas en un gran menú que funciona como los antiguos *Marking Menus* de *Power Animator*. Una vez abierto se puede trabajar sacando las herramientas según se vayan necesitando o, en el caso en el que se vaya a utilizar la misma herramienta repetidamente, se puede dejar fija de manera que no haga falta volver innecesariamente al menú central cada vez que se quisiera utilizar. Lo que se ha buscado con este método de trabajo es aumentar la rapidez, evitando tener que acudir una y otra vez a la paleta de herramientas y, por lo que comentó Javier, después de usarlo durante un corto período de tiempo los resultados son interesantes.

La SBD, *Scene Block Diagram*, que existía antes en Power Animator, ha crecido para transformarse en la función que lleva como nombre *HyperGraph*. En ella, la representación de los objetos que conforman una escena es mucho mas completa. Por ejemplo, el proceso de generación que ha tenido cada objeto está representado como un gráfico de dependencias donde se distingue con claridad

dicho proceso. En la demo se pudo ver otra aplicación de esta función. Bastó con acudir al editor de materiales y arrastrar una de las texturas hasta la ventana de *HyperGraph* para localizar a qué objeto estaba asignada. Para agrupar objetos en un mismo nodo, es suficiente con arrastrarlos al interior de esta ventana y colocarlos en la posición deseada.

Trabajar en un modelo con todas las texturas y materiales visibles, todas las luces de la escena activadas, en la ventana de perspectiva y sin que ello supusiera la eternización del proceso en aquellos equipos que lo soportaran, era del todo imposible. Pues bien, parece ser que ha cambiado. Antes, a lo máximo a lo que se podía aspirar era a trabajar en *TglShade* y, si teníamos al protagonista, a su familia y al vecino de enfrente, lo mejor era olvidarse de ello. Toda la demostración se hizo con texturas, luces etc., y la cosa funcionaba. También se presentó un nuevo y curioso modo de trabajo, *X-Ray*, que proporciona al modelo una transparencia parcial y que en ese momento se usó para seleccionar con facilidad los esqueletos asignados a Mel. Algo que complementa esto es la utilización de cálculos ya procesados consiguiendo que, por ejemplo, mientras transcurre un *Playback* de una animación, se pueda mover la cámara con facilidad dentro de la escena.

Hay algo más en esta carrera por la velocidad y optimización de proceso. *Switching* es un concepto muy conocido por todo aquel que esté en contacto con los sistemas que se usan en los simuladores de vuelo. Estos sistemas demandan por encima de todas las cosas actividad en tiempo real. El sistema funciona de manera que, según se va alejando la cámara, la geometría de los objetos se simplifica acelerando todo el proceso. Aparte de actuar sobre la malla, esta función se puede aplicar también en el render.

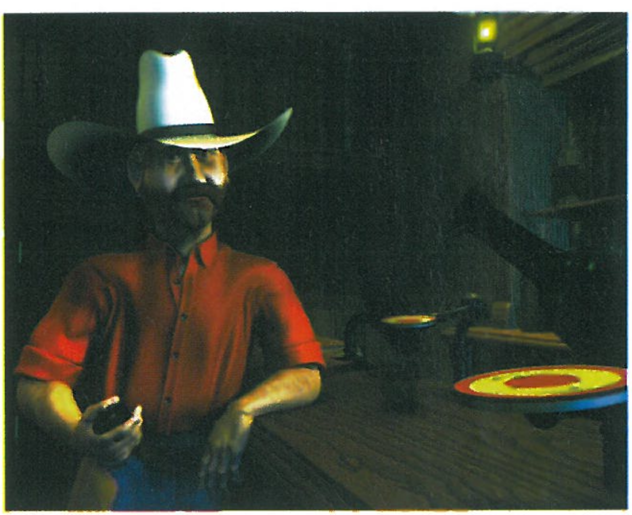
Lattice Box

Lattice Box es una caja de deformación de objetos que viene muy bien a la hora de transformar modelos con una geometría compleja donde es difícil discriminar zonas y nivel de influencia. Aunque no sea con este nombre, este tipo de aplicación existía ya en otros programas y su utilidad está fuera de toda duda. Lo que presenta Alias es una herramienta completísima, con un montón de opciones y un funcionamiento muy interactivo. En la demostración, *Lattice Box* se aplicó a varios componentes de Mel. En un momento determinado, la transformación que se le iba a practicar se hizo sacando los puntos de control de la caja de deformación, seleccionando algunos de ellos y determinando el grado de influencia para cada uno de ellos. Esto, en sí mismo, hace de *Lattice Box* una herramienta muy potente pero además, trabajando de esa manera, se pueden evitar las típicas arrugas que se producen en algunos casos cuando la deformación es muy pronunciada.

Flexors

Análogamente a lo que decía el párrafo anterior, trabajan los *Flexors*. Los *Flexors* consisten en una ayuda que se puede utilizar en objetos cuya animación está sujeta a la acción de unos esqueletos y en los que el movimiento que desarrollan produce en la malla efectos iguales a los que se producían en el caso del *Lattice Box*. Con ellos, el problema es muy sencillo de solucionar. El vaquero de nuestro cowboy se deformaba de una manera muy poco natural en la parte de la rodilla, por lo que bastó con colocarle un *Flexor* e ir afinando esta deformación en las distintas zonas, según fuera el caso.

En los últimos números de la revista se vienen tratando los menús del módulo de animación, por lo que todo el mundo que esté un poco al tanto recordará la ventana *Action Window*, desde la cual se controlaban las curvas de animación de los objetos de una escena. *Action Window* sigue estando ahí, aunque con una apariencia nueva. Esta vez sólo se usó para ver en qué modo afectaba el movimiento de uno de los *Keyframes* de la curva (creo que era de escalado en uno de los ejes) al modelo. Fue una pasada rápida por algo familiar, aunque no fuera la tónica que imperó durante la explicación .



Fotograma de la animación "El Saloon de Ruby", proyectada durante la presentación.

Skeletons

El tema de los esqueletos ocupó una parte importante. Los esqueletos en Alias basan su efectividad en una buena jerarquía de construcción y en la aplicación correcta de los elementos complementarios que existen en el programa.

Lattice Box es una caja de deformación de objetos que viene muy bien a la hora de transformar modelos con una geometría compleja donde es difícil discriminar zonas y nivel de influencia. Aunque no sea con este nombre, este tipo de aplicación existía ya en otros programas y su utilidad está fuera de toda duda.

Poder de modelado

Un ejemplo del poder y utilidad de Maya fue la manera con la que Javier Moreno creó rápidamente unos vaqueros que hubieran sido la envidia del departamento de diseño y estilo de Levi´s. Partiendo de unos cubos, fue tirando y empujando sobre la malla hasta conseguir que las primitivas pasaran a tener un aspecto realmente convincente. También con Artisan se puede actuar sobre propiedades dinámicas. Esto se pudo apreciar con calidad en el vuelo de la capa al que se le estaba aplicando viento. Con la brocha se iba determinando qué zonas debían de ser afectadas por el viento y en qué medida lo deberían hacer. La brocha trabajaba como los Brushes de Photoshop en cuanto a grosores y presión se refiere, todo muy sencillo e intuitivo. Como ya hemos dicho antes, las posibilidades de esta herramienta son verdaderamente impresionantes y colocan este software cada vez más arriba en cuanto a competitividad se refiere.



Simulación de pelo que Maya es capaz de conseguir.

La relación entre un objeto cualquiera y otro que es influenciado por éste aparece en Attribute Gral. Window. Aparte de esta función, en esta ventana de información están detalladas las distintas conexiones entre los modelos y sus componentes, de manera que es fácil ver cómo se comportarán éstos al dotarlos de movimiento.

Estamos hablando de *Constraints*, *Handles*, *IK Handles* y Límites. En Maya parece ser que esto no ha cambiado, o por lo menos de una manera radical. *Walk Window* es una función complementaria de las que acabamos de nombrar. Gracias a ella se puede matizar el movimiento que tenga un esqueleto partiendo de un movimiento base, pudiendo ser más dinámico, más pesado o con más o menos balanceo. Con unos *Keyfames* que propongan un comportamiento estándar en la acción de un esqueleto de lo que pudiera ser un humano y la acción de *Walk Window*, podemos tener a Elle MacPherson contoneándose en nuestro monitor o a Santiago Segura "arrastrándose pesadamente" hacia nosotros. Otro avance en este apartado, se ha proporcionado la posibilidad de trabajar con cadenas de esqueletos dobles, lo cual constituye una mejora más que sensible con la que se abre más su campo de funcionalidad.

Ventana de *Shaders* de Maya.

Mel entra al bar de Ruby y, tras una ojeada al lugar, pregunta quién va a servir un trago a un sediento cowboy como él. Este anecdótico apunte sirve para introducir el módulo de Control Facial y el Sincronizador de sonido. El Sincronizador de sonido relaciona de un modo visual el gráfico de, en este caso, la voz de Mel, con los gestos de su cara en el tiempo, lo que es muy útil a la hora de coordinar con exactitud sonido y movimiento. En el menú de Control Facial se pueden establecer con anterioridad los gestos que va a realizar el personaje, creándolos en el lenguaje de programación que usa Maya, MEL. Una vez hecho esto, se van colocando los *Keyframes* donde sea necesario, siendo suficiente mover los *displays* que acotan estos gestos para ver directamente los resultados sobre el modelo.

La relación entre un objeto cualquiera y otro que es influenciado por éste aparece en *Attribute Gral. Window*. Aparte de esta función, en esta ventana de información están detalladas las distintas conexiones entre los modelos y sus componentes, de manera que es fácil ver cómo se comportarán éstos al dotarlos de movimiento. Las dinámicas se han revisado y han avanzado para hacerlas más lógicas, más controlables y para que su comportamiento se acerque lo más posible al real. *Attribute Editor* se presenta como la opción desde la cual se han de manejar las partículas siguiendo la premisa de la que hablábamos en las líneas anteriores. El hecho de tener todas las dinámicas integradas en el software hace que se pueda determinar cuándo se quiere que dichas partículas entren en acción. Dentro de Maya, los objetos se pueden asociar a partículas, posibilitando que este objeto se transforme en partículas en un momento determinado. Como ejemplo, se pasaron unas imágenes donde unas balas que se disparaban y que chocaban contra una pared se convertían en partículas que simulaban pequeñas explosiones en el momento en el que éstas impactaban en la pared.

A propósito de esto, hay que hablar de una de las novedades mas importantes de todas las que aporta Maya. Nos estamos refiriendo a los *Soft Bodies*. Estos consisten en la posibilidad que hay de aplicar partículas a los puntos de control que tienen todas las superficies NURBS. La pared donde iban a parar las balas sufría físicamente el impacto, es decir, la transformación del plano que hacía de pared, se producía directamente sobre la malla. El plano era un *Soft Body* que desaparecía por aquellas partes donde hacían blanco los proyectiles. Otro de los ejemplos incidía un poco más en este sentido. La escena mostraba cómo al caminar un personaje sobre un cuerpo blando, éste se deformaba formando la silueta de las botas. Las aplicaciones de esta herramienta son interesantísimas, telas, líquidos, etc., por lo que, en mi opinión, son una de las estrellas de Maya.

Soft Bodies

Si los *Soft Bodies*, *Lattice Box*, MEL o el nuevo interfaz habían aparecido en el Broadcast y ya no nos pillaban de nuevas, el secreto que no se ha desvelado hasta ahora era Artisan. Debido a la cantidad de aplicaciones que tiene esta herramienta y al ínfimo contacto que hemos tenido con ella, es muy difícil establecer con criterio su campo de acción. Esencialmente, es una especie de esculpidor que vale tanto para producir deformaciones en la malla como para suavizar uniones, para seleccionar puntos de control como para ejecutar un *Script* en MEL situándolos con la brocha, e incluso se pueden pintar propiedades dinámicas y el nivel de influencia que éstas provocan en la geometría.

Por último, agradecer a Cristina Díaz, de SGO, su ayuda y su atención al realizar este reportaje.

Bruno de la Calva









POV RAY

Directivas condicionales y funciones

Nivel: **Avanzado**

Autor: **Enrique Urbaneja**

El POV-Team incorporó en la última versión de POV un conjunto de nuevas directivas en el lenguaje escénico del programa: las directivas condicionales y las funciones sobre objetos. Operaciones de gran utilidad para generar trayectorias o para construir figuras a partir de ecuaciones paramétricas, como las que han ilustrado este curso hasta el momento.

Actualmente estas directivas condicionales son ocho. Su estudio ocupará estas páginas durante varios meses aunque antes de ser estudiadas habrá que saber declarar identificadores, las variables con las que se controlarán estos procesos.

#DECLARE

Ésta es la directiva que permite declarar identificadores y su uso ya se estudió en los primeros números de este curso, aunque presenta cierta diferencia en cuanto a su uso, así es que se comentará de nuevo la sintaxis:

#declare IDENTIFICADOR = OBJETO

Donde *IDENTIFICADOR* es el nombre de la variable, siempre con menos de cuarenta caracteres, y *OBJETO* puede ser cualquiera de los objetos que se muestran en el cuadro 1.

En el cuadro 2 se declaran algunos a modo de ejemplo.

Atendiendo a estas declaraciones del cuadro 1, se podrían utilizar en un fichero escénico de la siguiente forma:

```
object {
 sphere { Pos, Pi }
 pigment { Rojo }
}
```

Sin embargo, el sistema de manipulación de identificadores en el lenguaje escénico no está muy desarrollado todavía, ya que cada vez que se desea cambiar alguna de las variables creadas, es necesaria la redeclaración de la misma.



Por ejemplo, si se declara una variable cuyo identificador sea Sumatorio: *#declare Sumatorio = 0* y se quiere incrementar su valor en uno, se debe realizar la siguiente operación: *#declare Sumatorio = Sumatorio + 1.*

#IFDEF & #IFNDEF

Puede darse el caso de que un objeto que declarásemos en nuestro fichero escénico estuviese ya declarado en algún fichero incluido en nuestro proyecto.

Con el propósito de detectar este tipo de situaciones, el POV-Team añadió estas dos directivas.

No son más que una versión de la directiva que se estudiará a continuación *#IF ... #ELSE*, y la sintaxis es la siguiente:

#ifdef (OBJETO)

Aquí se entraría en caso de que se detectara un objeto declarado anteriormente y de iguales propiedades a OBJETO.

Si se desea, se puede utilizar el declarado anteriormente con la siguiente instrucción:

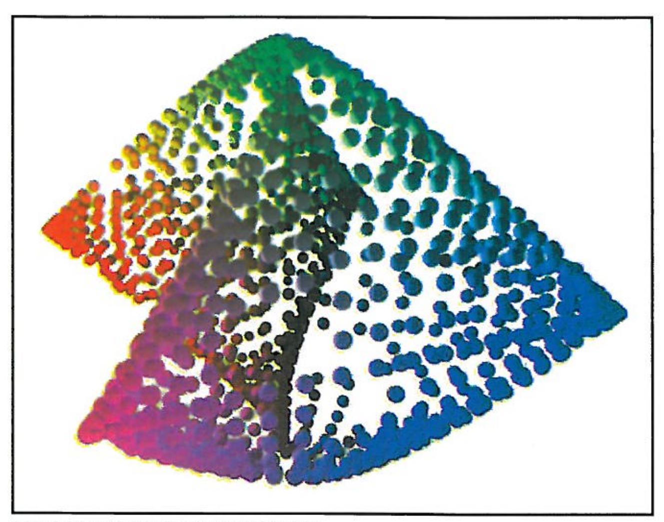
```
  object { OBJETO }
#else
```

Y aquí se entraría si no estuviera declarado, en donde se declara:

```
  object { ... }
#end
```

#ifndef presenta una estructura equivalente aunque su funcionamiento es el contrario, preguntando si un objeto determinado "no ha sido declarado" hasta el momento.

Y antes de pasar a las siguientes directivas, decir que la directiva *#else* es opcional y que la instanciación de estas directivas surgirá efecto siempre y cuando se localicen al principio de nuestro fichero escénico.



Ejemplo de ecuación paramétrica.

Simple imagen realizada a partir de esferas.

#IF & #ELSE

Estas directivas son las más simples del lenguaje de programación interno de POV, y su funcionamiento es exactamente igual al que tiene en cualquier lenguaje de programación.

La estructura *if-else* cuenta con tres parámetros, *#if, #else y #end*, y la estructura es la siguiente:

```
#if ( Cond )
        sentencias if
#else
        sentencias else
#end
```

Si se cumple la condición, se ejecutarán las sentencias que POV encuentre hasta encontrar la directiva *#else* y, en el caso de que no se cumpla la condición, se ejecutarán las sentencias que se encuentren desde la directiva *#else* hasta la directiva *#end*, que marcará el final de la estructura.

Sin embargo, la directiva *#else* es opcional, ya que nos puede interesar realizar algo cuando el valor de la condición sea el que nosotros queramos y, en caso de que no ocurra así, no hacer nada.

De todas formas, se utilice o no la directiva *#else, #end* es requerida siempre para finalizar la estructura condicional.

La condición será una condición booleana, es decir, verdadero o falso; el valor falso vendrá determinado por el valor numérico 0.0, mientras que cualquier otro valor distinto de éste será considerado como valor booleano igual a verdadero.

Y mucho cuidado con los paréntesis que encierra la condición ya que su utilización es obligatoria.

La siguiente estructura también pertenece al conjunto de las estructuras condicionales: Caso: Acción, muy útil cuando tenemos bastantes casos en los que tenemos que actuar de formas distintas en cada uno de ellos.

#SWITCH, #CASE & #RANGE

```
#switch ( Valor )

 #case (  Valor para el Caso1 )
    // Sentencias 1
 #break

 #case (  Valor para el Caso2 )
    // Sentencias 2
 #break

 #range ( MÍNIMO1, MÁXIMO1 )
    // Sentencias 3
 #break

 #range ( MÍNIMO2, MÁXIMO3 )
    // Sentencias 4
 #break

 #else
    // Sentencias 5
#end
```

Esta estructura posee seis palabras clave: *#switch, #case, #break, #range, #else y #end.*

El funcionamiento es el siguiente: dependiendo del valor que le entre a *#switch* se irán evaluando los casos que se encuentren en el cuerpo de la estructura hasta que se encuentre la directiva *#end,* o se entre en algún conjunto de sentencias porque el valor se corresponda a uno de los valores de algún *#case*.

Supongamos que Valor es igual a cinco, y que el valor para el Caso1 vale cuatro y el valor para el Caso2 es cinco.

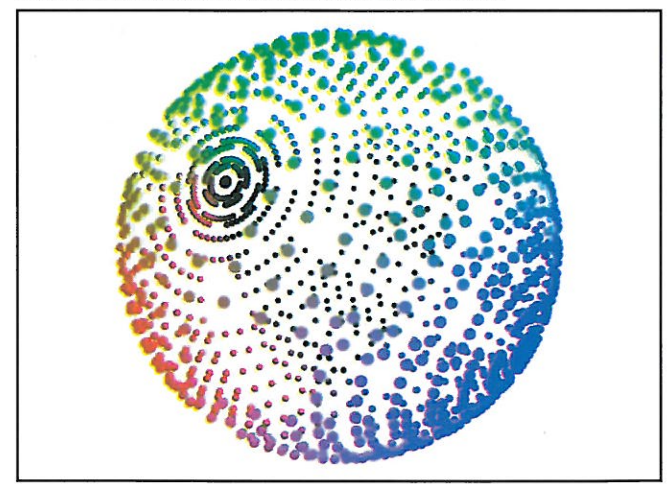
En el momento en que POV compila el código y encuentra una estructura de este tipo, realiza la evaluación y, en este caso, tras comprobar que no el valor de *switch* no se corresponde con el del caso1, saltaría al de caso2.

Al llegar a este punto, y comprobar que el valor del caso es el mismo que el de evaluación, es decir, el del *switch,* POV ejecu-

Otro claro ejemplo de ecuación paramétrica.



Esfera creada a partir de ecuaciones paramétricas.



CUADRO 1. OBJETOS PERMITIDOS EN LA DECLARACIÓN DE UN IDENTIFICADOR

float	*pigment*	*pigment_map*	*atmosphere*
vector	*normal*	*slope_map*	*fog*
color	*finish*	*normal_map*	*rainbow*
string	*halo*	*camera*	*sky_sphere*
texture	*color_map*	*light_source*	*transform*

CUADRO 2. DECLARACIÓN DE IDENTIFICADORES

```
#declare Pi = 3.141592654
#declare Pos = < 2 1 4 >
#declare Rojo = color rgbf < 1 0 0 0 >
#declare miFuente = "courier.ttf"
#declare miTextura = texture { pigment { color Rojo } }
#declare miPigment = pigment { checker Blanco, Negro }
#declare miNormal = normal { bumps .2 }
#declare miFinish = finish { ambient .2 diffuse .5 }
#declare miHalo = halo { emitting ... }
#declare unColor_Map = color_map { [0 1 Rojo Verde ] }
```

taría las sentencias que se encontraran en su cuerpo hasta la directiva *#break*, en este caso Sentencias 2 y, finalmente, saldría de la estructura *switch*.

Hasta aquí, y como habrán podido comprobar los lectores con conocimientos del lenguaje de programación C, no hay variación alguna con respecto a la estructura equivalente en este lenguaje, exceptuando la sintaxis. Sin embargo, existe una diferencia considerable: en una estructura de este tipo en el lenguaje C no se pueden especificar rangos de valores, siempre hay que especificar un valor fijo para cada *case*, defecto que no posee el lenguaje Pascal, en el que sí se pueden especificar rangos de valores.

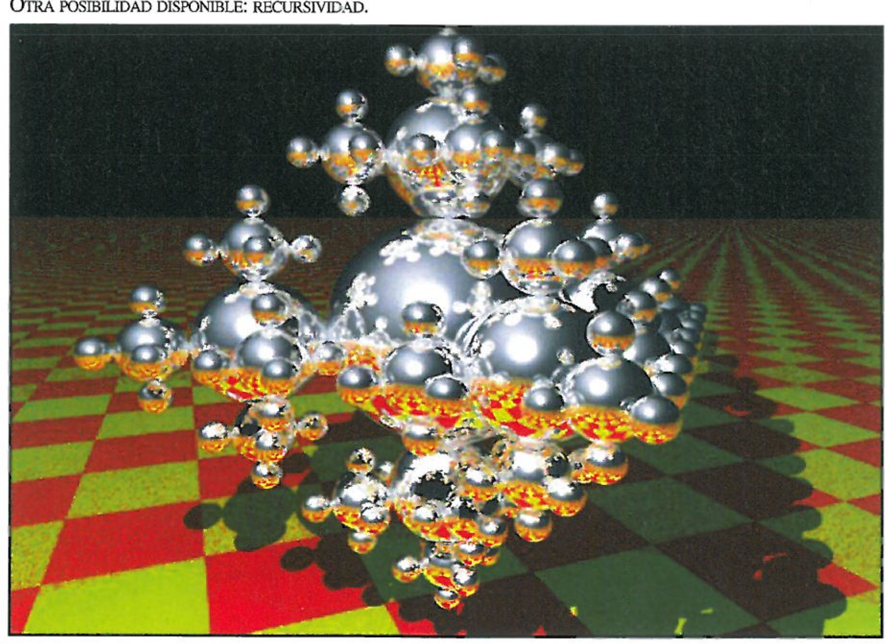
Como se puede ver, el POV-Team ha pensado en todo a la hora de realizar este mini-lenguaje de programación, cogiendo lo mejor de los más populares lenguajes.

De esta forma, además de valores fijos, se pueden especificar rangos de valores de la siguiente forma: *#range (valor mínimo, valor máximo)*, en donde valor mínimo representa al *valor mínimo* del rango de valores y *valor máximo* al máximo.

Las sentencias que se encuentren entre esta directiva y la palabra clave *#break* se llevarán a cabo si se cumple que el valor de entrada sea mayor o igual al valor mínimo, y en menor o igual al valor máximo.

Y, por último, tenemos la posibilidad de ejecutar otras sentencias en el caso de que el valor introducido en el *#swicht* no se haya correspondido con ninguno de los casos y no se haya encontrado entre ninguno de los rangos especificados, la forma es utilizando la directiva *#else* seguido de las sentencias pertinentes y finalizando con *#end*, terminándose, así, además la estructura entera.

OTRA POSIBILIDAD DISPONIBLE: RECURSIVIDAD.



Y una nota, si después de algún *#case* o *#range* y sus respectivas sentencias no se cierta el cuerpo con *#break*, se saltará tras su ejecución a la siguiente estructura, bien sea ésta un *#case* o *#range*.

#WHILE

La estructura *#while* permite iterar un proceso hasta que una condición, que declaramos nosotros, se deje de cumplir.

La sintaxis es la siguiente:

```
#while (cond )
 proceso1
 proceso2
...
procesoN
#end
```

Este bucle de recorrido variable, que también es como son llamados, realizaría los N procesos tantas veces como la variable *cond* tuviera un valor distinto de cero. Sin embargo, un bucle *#while* se puede convertir en un bucle de recorrido fijo. El siguiente código sirve de ejemplo:

```
#declare cont = 5
#while (cont > 0)
 #declare cont = cont – 1
 procesos
#end
```

En cada entrada al bucle se realizan dos tareas; la primera, decrementar en uno el valor de la variable *cont*, cuyo valor inicial es 5 y, la segunda, un conjunto de procesos cualquiera.

Calcular el número de iteraciones que realizará el bucle *#while* es bastante secncillo: si inicialmente el contador tiene el valor 5, y en cada iteración se decrementa en uno, el número de veces que se realizará el proceso será cinco.

Una vez estudiados los identificadores y las directivas condicionales, queda por estudiar las funciones sobre objetos: sobre *floats*, *vectores* y *strings*:

FUNCIONES SOBRE *FLOATS*

POV incorpora, desde la versión 3.0, las funciones imprescindibles para realizar cálculos matemáticos como los que se necesitan, por ejemplo, en la generación de superficies paramétricas, en donde se utilizan coordenadas polares, y funciones como el coseno o el seno de un ángulo son necesarias.

El cuadro 3 recoge este conjunto de operaciones ordenadas por orden alfabético y que con conocimientos básicos de matemáticas pueden ser reconocidas fácilmente.

Por el momento no se puede más que comentar cada una de ellas, emplazándoos para el mes que viene en donde estudiaremos algunos ejemplos prácticos.

CUADRO 3. FUNCIONES SOBRE FLOATS

abs (A)	*ceil(A)*	*floor(A)*	*min(A,B)*
acos(A)	*cos(A)*	*int(A)*	*mod(A,B)radians(A)*
asin(A)	*degrees(A)*	*log(A)*	*rand(A)*
atan2(A,B)	*exp(A)*	*max(A,B)*	*seed(A)*

Empecemos, pues, por *abs (A)*; implementada como función, devuelve el valor absoluto de *A*, así, por ejemplo, si queremos que una variable pase a tomar el valor absoluto de ella misma, la operación sería la siguiente: *#declare Var = abs(Var)*.

acos(A) devuelve el ángulo del coseno de un ángulo, representado por A, y al igual pero hablando del seno, *asin(A)* retorna el ángulo del seno de un ángulo, representado por A.

atan2(A,B) devuelve el ángulo cuya tangente es A/B, *ceil(A)* devolverá el menor número entero mayor que A, *cos(A)* el coseno de A, *degrees(A)* convierte de radianes a grados, y como también es una función devuelve el resultado.

exp(A) devuelve el exponencial de A, *floor(A)* devolverá el mayor número entero menor que A, *int(A)* la parte entera de A, *log(A)* el logaritmo de A, *max(A,B)* el máximo entre A y B, es decir, devuelve A si A > B, B si B > A y A si A es igual a B, *min(A,B)* el mínimo de los dos.

mod(A,B) = A[B], *radians(A)* devuelve un valor en radianes a partir de otro valor: A, que se encuentre en grados, *rand(A)* devuelve un valor aleatorio donde A representa el mayor número que puede llegar a generar aleatoriamente esta función, el menor no se requiere ya que es, por defecto, 0.

Y para que esta última función se ejecute, es necesario llamar a la función *seed(A)*, que devuelve una semilla a partir de un valor: A.

Breves ejemplos antes de acabar por este mes para fijar cómo se deben utilizar en asignaciones:

#declare Beta = 90
#declare Coseno = cos(Real)

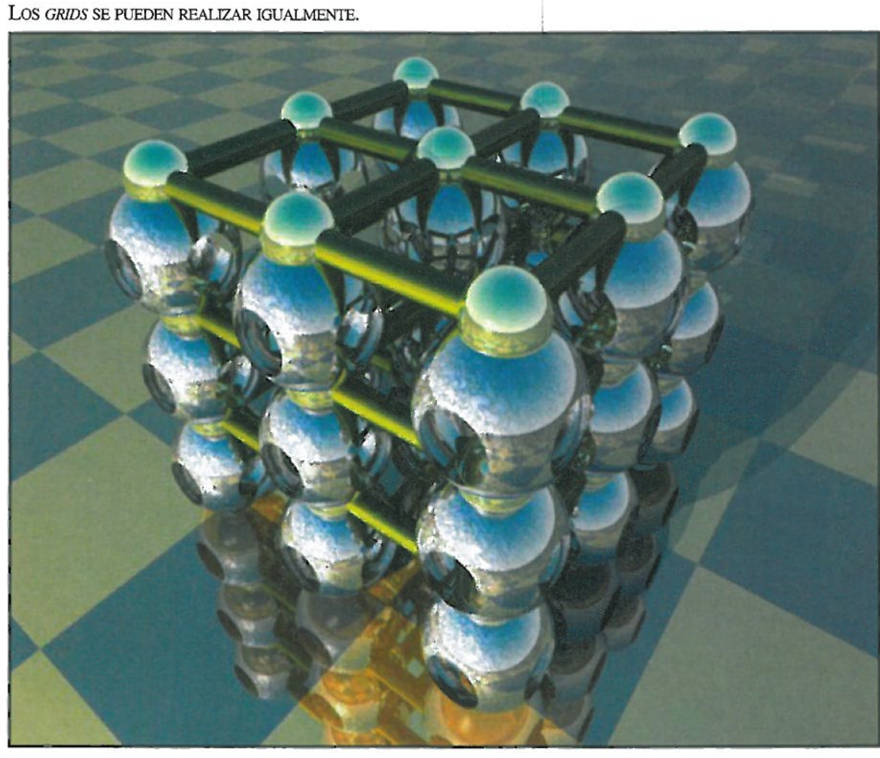
De esta forma, el valor de Coseno será el coseno de 90º, es decir, cero.

FUNCIONES SOBRE VECTORES

Las siguientes funciones permiten, a partir de uno o más vectores y valores en formato *float*, calcular operaciones como rotaciones, productos escalares etc... y todas ellas funcionan, por defecto, y únicamente con vectores de tres coordenadas.

La notación sigue estando en inglés como vais a poder comprobar:

vaxis_rotate(A,B,F), donde A y B son vectores, y F es un escalar que representa el ángulo a rotar el vector A, tomando como eje de rotación la dirección del vector B.


LOS *GRIDS* SE PUEDEN REALIZAR IGUALMENTE.


UN EJEMPLO DE PARAMETRIZACIÓN IMPLÍCITA.

v_cross(A,B) es el equivalente al producto vectorial entre dos vectores, es decir, devuelve un vector perpendicular a estos dos.

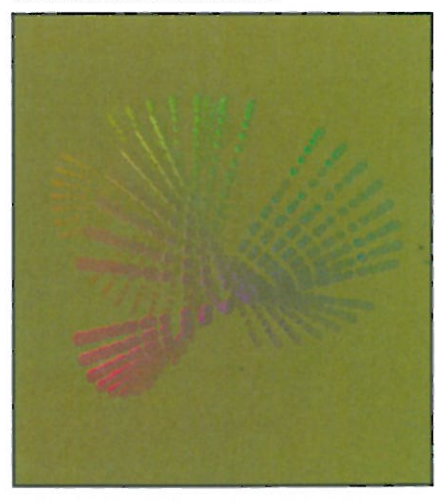
v_dot(A,B): el producto escalar entre dos vectores, y retorna un *float*.

El módulo de un vector se consigue con la función *vlength(A)* y puede ser utilizado, por ejemplo, para calcular la longitud entre dos puntos, creando un vector con ellos dos, y, posteriormente, con esta función hallando la distancia que los separa.

Cuando nos interese que un vector tenga módulo 1, es decir, que sea un vector unitario, se podrá utilizar la función *vnormalize(A)*, que devolverá el vector normalizado.

vrotate(A,B) por su parte, rota un vector A tomando como origen el vector B.

Como se puede comprobar, no son muchas pero no falta ninguna función que no pueda ser construida a partir de éstas.

REPRESENTACIÓN PARAMÉTRICA.


CUADRO 4. FUNCIONES SOBRE VECTORES

vaxis_rotate(A,B,F)
v_cross(A,B)
v_dot(A,B)
vlength(A)
vnormalize(A)
vrotate(A,B)



CURSO

CALIGARI TRUE SPACE

PC

Las funciones de animación
Autor: César M. Vicente

Nivel: Medio

En el presente artículo se comenzarán a mostrar, con los ejemplos prácticos necesarios, las funciones principales de animación que nos permitirán realizar la secuencia de la pelea de motos de luz de la película.

En anteriores artículos se ha ido viendo cómo se construía el modelo para la futura animación, así como un pequeño escenario para ésta, y se le han dado materiales a los objetos explicando cómo modificar y aplicar de manera más eficiente y rápida, por lo que sólo queda por mostrar todo lo necesario sobre los controles y parámetros de animación, aspecto obligatorio para lograr una secuencia de las motos de luz persiguiéndose por la rejilla de juegos.

LOS CONTROLES

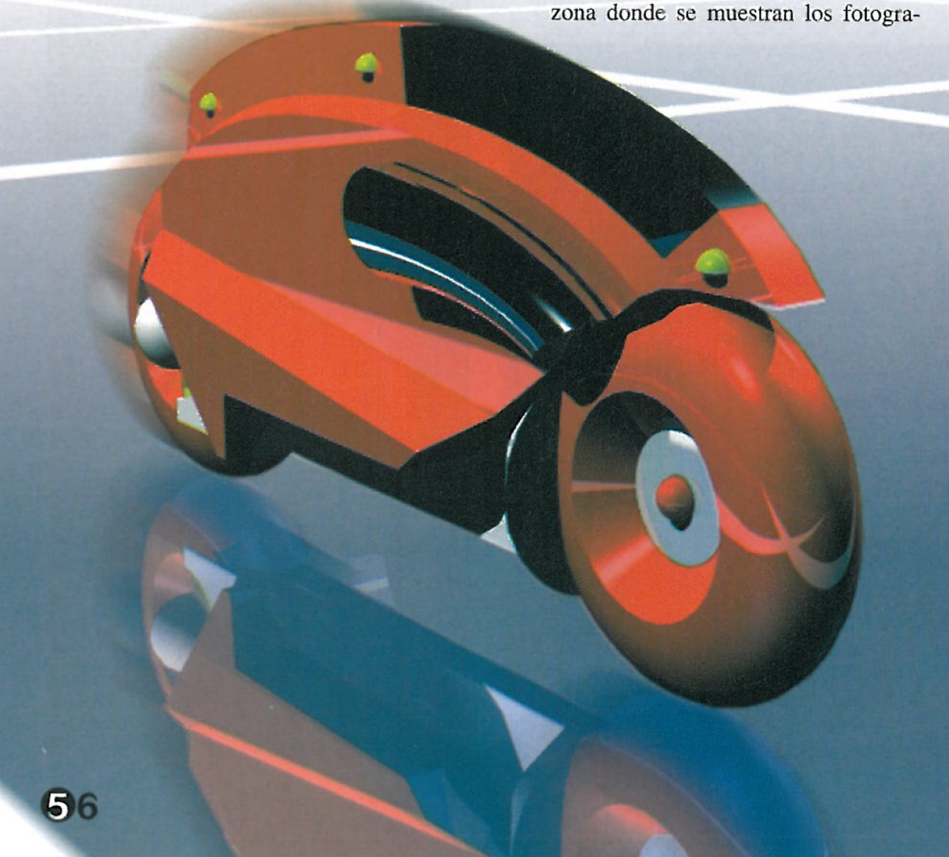
Para realizar una animación el programa dispone de unos controles bastante simples, pero con una funcionalidad muy alta, a la par que sencilla de manejar. Cuando se activa el icono dedicado a realizar la siguiente función: aparece en pantalla una ventana donde se encuentran los controles de la animación (Cuadro 1).

Con estos controles, realizar todas las funciones de animación se convierte en algo sencillo y, para comprobarlo, una muestra simple: un ejemplo donde se nos demuestre cómo funciona.

Con el plano del escenario visto desde la vista *Top*, se selecciona la moto en una posición inicial. Se ha escogido la vista *Top* porque, por ahora, todo el movimiento de las motos se va a realizar en el plano horizontal, sin cambiar de altura, por lo que es la vista ideal para realizar este tipo de animación.

Una vez seleccionada la moto, se activa el icono de animación y se colocan todas las piezas en la posición inicial que se quiera. Ahora, basta con poner sobre la zona donde se muestran los fotogramas (imagen fija perteneciente a una animación o fotograma), en el número que se quiere para realizar esta animación, por ejemplo, en 30.

Basta ahora con mover los objetos a la posición final donde estarán en este fotograma, y al pulsar el botón de *Play* del controlador se apreciará el cambio de posición.



LA PRECISIÓN

Es posible que realizar una animación sea el trabajo que más paciencia y atención exija como parte del conjunto que forma el trabajo en 3D, por lo que disponer de la máxima información posible de cómo se está creando ésta, los parámetros de control, así como un acceso fácil a los puntos de control de esta animación (KEYS) es fundamental para trabajar de una forma fácil y fluida.

Algunos programas se complican de gran manera al mostrar estas funciones y otros, como el Softimage o el 3D Studio MAX (uno de los mejores en lo que es el control de una animación), tienen tal cantidad de datos y posibilidades que en muchos casos avasallan y agobian a todos los que, por primera vez, se enfrentan a esta situación.

Una de las mejores características que posee el Caligari es que, a pesar de no tener un excesivo número de funciones de animación (por lo menos en lo que se refiere a programa estándar, ya que existen gran cantidad de funciones en forma de plug-in´s para añadirle), es que las que tiene son de muy fácil acceso y aprendizaje, y que, sobre todo, cumplen con sus funciones de una manera espléndida, permitiendo a los usuarios noveles el entrar y practicar las funciones básicas de animación en muy poco tiempo.

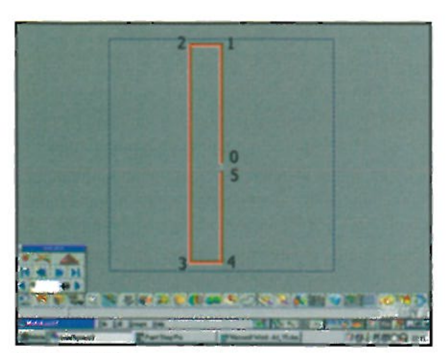


FIGURA 2. EN ESTE EJEMPLO SE PUEDE VER EL DESPLAZAMIENTO QUE SEGUIRÁ LA MOTO, SEGÚN LOS PUNTOS DE CONTROL.

UNA ANIMACIÓN SENCILLA

Se va a realizar una pequeña animación con la moto de luz, con una cámara fija, pero que va a seguir el movimiento de la moto en sus correrías por toda la rejilla de juegos, pero sin la estela (eso vendrá después).

Lo primero es presentar lo que se va hacer, fundamental para no dejar nada a la improvisación, culpable en muchos casos de grandes pérdidas de tiempo y, fundamentalmente, de malos resultados.

El planteamiento que se hará con la moto es el que se muestra en la figura 2, con una raya de color rojo marcando todas las trayectorias que va a seguir la moto. Como se puede ver, la posición final corresponde con la inicial, así, cuando se vaya a sacar en formato AVI se podrá poner en modo cíclico.

Como se aprecia en la figura 2, el movimiento de la moto va a ser, partiendo desde el punto 0, seguir los puntos 1, 2, 3, 4 y 5, que será el mismo que el de la posición de partida inicial de la moto en el punto 0.

Un truco antes de iniciar la animación consiste en colocar unos cubos, justo en el sitio donde se producirá los controles de la animación y luego traspasarle estos parámetros de posición a la figura que se va a animar, en este caso la moto.

LOS CONTROLES

Los controles que presenta la animación vienen resumidos en esta ventana y sus funciones son (de izquierda a derecha y de arriba abajo):

REC: Graba todas las modificaciones que se realicen en la pantalla, según el fotograma en que se esté. Si se pulsa en las opciones internas, aparece un cuadro de diálogo en el que se muestra la opción de *autoshave*, para que detecte automáticamente todas las modificaciones y las salve en el fotograma activo.

KEYS/UNKEYS: Activa o borra un punto de control de la animación (*Key*). Es interesante dominar el control de esta función ya que determina, en muchas ocasiones, el perfecto control de una animación.

PLAY: Ejecuta una animación, en el tiempo y velocidad predefinidos, según el formato elegido de salida de la animación PAL/SECAM, FILM, o NTSC.

IR AL PRINCIPIO: Localiza el puntero de localización de la animación al principio de ésta.

IR AL KEY ANTERIOR: Retrocede hasta el anterior *Key* de control del objeto seleccionado. Si se tiene activada la animación para toda la escena, retrocede hasta el *Key* más próximo de cualquier objeto.



IR AL KEY SIGUIENTE: Igual que el anterior, pero avanzando hasta el siguiente *Key*.

IR AL FINAL: Se coloca el *Key* de control último de la animación.

RETROCEDER UN FRAME: Pasa al fotograma anterior.

CUADRO DE FRAME: Muestra el fotograma activo. Aquí también se le puede introducir un número directamente, para ir al fotograma que se quiera, y así añadir un nuevo Key de control para un objeto.

AVANZAR UN FRAME: Avanza un solo fotograma hacia delante.

Todas las teclas que tienen control sobre el avance o retroceso entre los fotogramas y los puntos de control, tienen un submenú de opciones para saber qué controles tiene activos en cada momento.

LOS TIEMPOS DE ANIMACIÓN

Es uno de los aspectos sobre el que hay que tener más control en la animación y se trata del lugar y el momento en que estará cada pieza en la animación. Esto, que en un primer instante puede parecer sencillo, es realmente una de las mayores complicaciones que se presentan al animar, sobre todo si son complejas, y hay que sincronizarlas con sonidos de acompañamiento (en cuanto a esto último, el programa dispone de un novedoso sistema de efectos especiales asociado a los objetos de denominado RSX 3D que permite incorporar y variar según la posición de la cámara un sonido asociado a un objeto y variarlo estereofónicamente y con efectos, a la vez que varía la imagen, pero esto ya se verá en futuros capítulos).

Lo primero que hay que averiguar es para qué formato se va a sacar la animación, ya que la situación de los fotogramas variará de manera importante según sea el formato de salida.

FORMATOS DE SALIDA

En principio existen tres formatos de televisión y cine que son estándar a la hora de trabajar:

• *NTSC*, sistema de televisión que existe en Estados Unidos y en Japón que tiene,

FIGURA 1. LOS CONTROLES DE ANIMACIÓN QUE PRESENTA EL PROGRAMA SON BASTANTE SIMPLES PERO MUY FUNCIONALES.

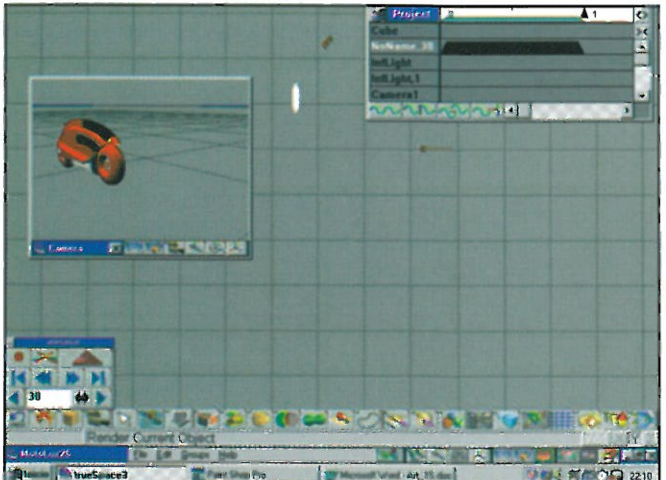
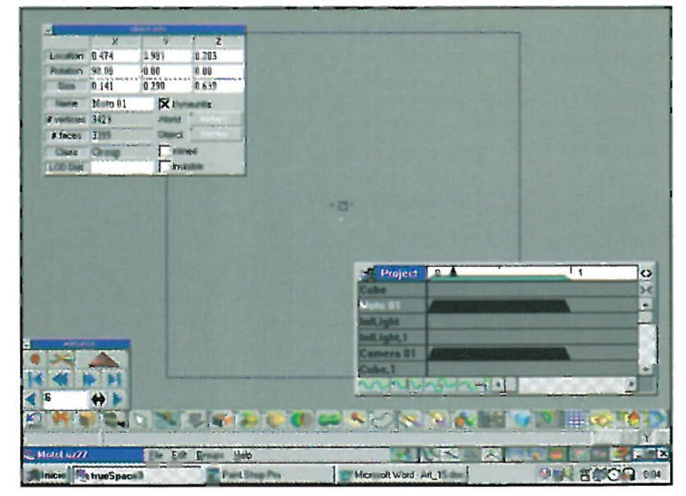


FIGURA 3. SI RESULTA DIFÍCIL DESIGNAR Y SELECCIONAR LOS OBJETIVOS, SE PUEDE UTILIZAR LA VENTANA DE CONTROL DE PROYECTOS Y SELECCIONARLOS DESDE AHÍ.



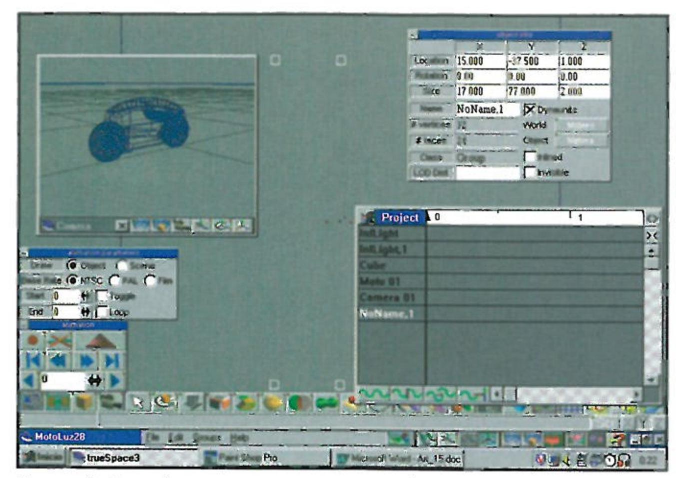


FIGURA 4. SE SITÚAN LOS CUBOS QUE FORMARÁN PARTE DEL SISTEMA DE CONTROL DE LA ANIMACIÓN, PARA LUEGO PASARLE LOS PARÁMETROS A LA MOTO.

FIGURA 5. SE COLOCA LA MOTO EN LA POSICIÓN DE LOS CUBOS, INTENTANDO SITUARSE EN EL FOTOGRAMA PRECISO, Y PONER LA MOTO EN EL PUNTO DE GIRO.

como característica principal, una altura de 525 líneas de resolución y una frecuencia de barrido de 60 Hz, por lo que, como cada imagen necesita 2 barridos completos para generarse, necesita 30 fotogramas para generar un segundo de vídeo.

• *PAL*, sistema desarrollado por Philips y que, actualmente, se emplea en toda Europa a excepción de Francia, el cual utiliza el Secam, variante del anterior, con modificaciones sobre la información del color. La resolución vertical es 625 líneas y la frecuencia de refresco es de 50 Hz, por lo que, con el sistema mencionado anteriormente, se necesitan 25 fotogramas para generar un segundo de animación.

• *Film* es el sistema de cine actual y necesita 24 fotogramas para generar un segundo de animación.

De todas las maneras, esto es información interior, y si no se tiene pensado sacarlo a vídeo, y dejarlo en formato AVI, se pueden poner el número de fotogramas que se desee, aunque no es muy recomendable bajar de 15 como mínimo.

CÁLCULOS DE TIEMPO

Una vez indicado el tipo de sistema (en este caso Film), se pasará a calcular cuántos fotogramas hacen falta para toda la animación.

El sistema siempre dependerá de la longitud entre los puntos, y del tiempo que queremos que dure. En principio, se va a suponer que va a durar 12 segundos (luego todo esto es modificable con la ventana de seguimiento de proyectos, ver cuadro 2). Entonces hay que repartir esos 12 segundos entre toda la animación. Mirándola se puede observar lo siguiente: existen dos rectas verticales que son unas 5 veces más grandes que las horizontales, por lo que para que la moto tenga siempre una velocidad constante (de las aceleraciones se hablará en próximos números), habrá que repartir todos los fotogramas de una manera equitativa manteniendo estas proporciones, por lo que a la recta entre 0 y 1 le corresponderán 2´5 segundos (24x2+12 fotogramas = 60 fotogramas), entre 1 y 2, un segundo (24 fotogramas), entre 2 y 3, cinco segundos (120 fotogramas), entre 3 y 4, igual que entre 1 y 2, y entre 4 y 5, igual que entre 0 y 1.

Una vez calculados los tiempos en fotogramas, se ve que los puntos de control estarán situados en los fotogramas: 60, 84, 204, 228, y 276, el cual será el último fotograma de la animación.

Ahora, para realizar ésta se hace lo siguiente: se sitúa la moto en la posición de partida, en el fotograma 0, se apuntan las coordenadas de posición (importante para, posteriormente, devolverla al mismo sitio en el fotograma 276). Se pone el número del fotograma siguiente (60) y se sitúa la moto moviéndola en el eje vertical hasta la posición prefijada.

Una vez ahí, se sitúa el fotograma 61 y se gira 90 grados hacia la izquierda (se gasta un fotograma en hacer que gire); ahora se marca el fotograma 84, y se coloca en el punto 2, se aumenta un fotograma y se gira 90 grados hacia abajo, etc., así con todos los puntos de control.

Todo esto se puede hacer a mano, pero el resultado no será el mismo que si se hace teniendo la ventana de coordenadas abierta y poniendo sobre ella la posición exacta del movimiento de la moto.

Para terminar, y una vez comprobado que la moto hace exactamente lo que se quiere, se puede incorporar el movimiento de giro de la cámara para seguir a la moto a lo largo de toda la trayectoria. Para ello, basta con irse colocando en cada fotograma de cambio de punto de control de la animación (los marcados anteriormente), y girar la cámara para que apunte a la moto.

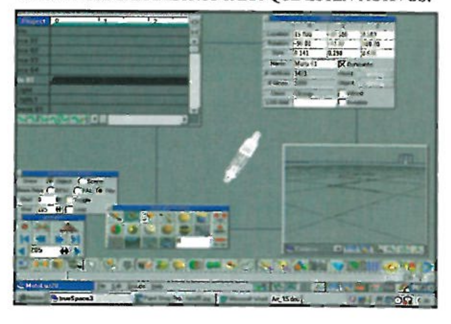
Lo mejor para hacer esto es dejar una ventana abierta con la cámara como encuadre, para ir viendo todo el proceso de la animación, y cómo quedará en el vídeo.

CONCLUSIÓN

Ahora que ya sabemos cómo se puede animar, sólo un poco, se puede empezar a realizar la animación anunciada en los números anteriores, mostrándose las estelas que dejan las motos, así como algún otro efecto sobre ellas.

El mes que viene, además del propio artículo, se incluirá en el CD Rom de la revista esta animación, con un escenario previo para que se pueda practicar todo lo visto hasta ahora. Así que, hasta el mes que viene.

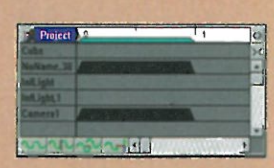
FIGURA 6. LOS FOTOGRAMAS SON INDEPENDIENTES ENTRE SÍ, SEGÚN PERTENEZCAN A UN GIRO, DESPLAZAMIENTO, ESCALADO. HABRÁ QUE ESTAR EN CADA UNO DE LOS PARÁMETROS PARA QUE ESTÉN ACTIVOS.



LA VENTANA DEL PROYECTO DE ANIMACIÓN

Esta ventana, que no ha variado mucho en las tres versiones del programa, permite un control sobre las secuencias de los diversos objetos que se encuentran en pantalla para una animación.

Asimismo, posibilita la visualización en tiempo por segundos o por fotogramas y muestra la situación con una barra de tiempos de los bloques en que interviene o se modifica un determinado objeto en pantalla.



El icono superior de la pantalla puede controlar también el momento de la animación, así como los iconos inferiores pueden incluir una serie completa de parámetros de repetición de una acción, sin tenerlo que hacer físicamente sobre la animación (ruedas y toda clase de objetos giratorios).

El programa profesional de modelado

MANUAL TÉCNICO LIGHTWAVE

Lightwave se ha convertido, desde su aparición, en una herramienta de modelado y animación en 3D a tener muy en cuenta. Su potencia ha sido demostrada en multitud de campos, incluyendo efectos especiales para las series de televisión "SeaQuest", "Babylon 5" o "Expediente X". Además, su disponibilidad en todas las plataformas existentes como Macintosh, Silicon Graphics, ALPHA y Amiga hace de esta herramienta la más estandarizada dentro del mundo de las 3D.

Esta obra introduce al lector en el mundo del modelado con Lightwave de una forma sencilla, práctica y amena, desde su filosofía de trabajo, el modelador o el editor de escenarios hasta las opciones más avanzadas como la creación de efectos atmosféricos, realización de animaciones, la dinámica del movimiento o el modelado orgánico. Todo ello con ejemplos prácticos para entender mejor cada uno de los conceptos.



INCLUYE:

- Estructura del paquete
- La interfaz de usuario
- Los menús del modelador
- Rótulos en 3D
- Esferas de colores
- Trabajo con capas
- Modelos orgánicos
- El editor de escenarios
- Los menús del editor
- Elementos de la escena
- Cómo añadir elementos a la escena
- Claves de animación
- La dinámica del movimiento
- Integración de elementos
- Cómo trabajar a diferentes escalas
- Efectos atmosféricos
- Luces y nieblas
- Movimiento difuminado
- Generación de vídeo
- Atajos de Teclado (Hot Keys)
- Recursos, módulos y tutoriales. Dónde buscarlos
- OpenGL, RenderGL, y QuickDraw

EN EL CD-ROM... LIGHTWAVE

- Directorio 3DFONTS. En este directorio encontraremos 231 fuentes 3D de letra para utilizar con Lightwave y dar a los textos una apariencia más profesional.
- Directorio ANIMS. El directorio ANIMS contiene unas animaciones en las que podremos apreciar la utilización de algunas de las demos de Plug-ins incluidas en el CD.
- Directorio BACKGROU. Conjunto de 15 imágenes de fondo (Backgrounds) para utilizar en nuestras escenas.
- Directorio DEMOS. Contiene demos de algunos de los Plug-ins más conocidos, como Particle Storm, Impact, Spiral Path Generator, Shell o VertiElectric.
- Directorio ESCENAS. En este directorio encontraremos escenas ya creadas con Lightwave, las cuales podremos renderizar para ver sus resultados o modificar con nuestros propios objetos para cambiar sus aspecto.
- Directorio OBJETOS. Una amplia colección de objetos 3D con más de 200 modelos para crear nuestras propias escenas. Entre estos modelos podemos encontrar todo lo que imaginemos: Vehículos, animales, naves de las series Star Wars, Star Trek y Babylon 5, modelos humanos, objetos. En definitiva, todo lo que nuestra mente pueda imaginar.
- Directorio PLUGINS
87 Plug-ins para Lightwave con los que conseguir diferentes efectos y resultados. Creación y edición de primitivas, modificación de Splines y NURBS o efectos especiales, entre otros.
- Directorio TEXTURAS
1896 texturas para envolver nuestros objetos o para utilizar como fondos. Las hay de todos los tipos imaginables, desde aguas hasta granitos, pasando por ladrillos, mármoles, metales, minerales, rocas, estucados, maderas, mosaicos o nubes.

RESERVA TU EJEMPLAR EN EL QUIOSCO ANTES DE QUE SE AGOTE.

OFERTA DE LANZAMIENTO: LIBRO + CD-ROM POR SÓLO 2.995 ptas.

Edita:



Solicite su ejemplar enviando este cupón por correo, por Fax: (91) 304.17.97 o llamando al teléfono (91) 304.06.22 de 9:00 a 19:00h.

Deseo que me envíen:
☐ MANUAL TÉCNICO LIGHTWAVE por 2995 + 450 ptas. gastos de envío.
☐ MANUAL TÉCNICO DE 3D STUDIO 4 por 2995 + 450 ptas. gastos de envío.
☐ MANUAL TÉCNICO DE COREL DRAW 7 por 2995 + 450 ptas. gastos de envío.
☐ DOS LIBROS POR SÓLO 4995 + 500 ptas. gastos de envío. (Indica con una cruz los dos)
☐ LOS TRES LIBROS POR SÓLO 6995 + 500 ptas. gastos de envío.

Nombre y apellidos .... Domicilio .... Población ....
Provincia .... CP .... Fecha de nacimiento .... DNI/NIF ....

FORMA DE PAGO

☐ Talón a PRENSA TÉCNICA ☐ Contra-reembolso   Firma,
☐ Giro postal nº .... de fecha ....
☐ Tarjeta de crédito ☐ VISA nº ☐ AMERICAN EXPRESS nº ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐
☐ Fecha de caducidad de la tarjeta .... Nombre del titular, si es distinto ....

Rellena este cupón y envíalo a:
PRENSA TÉCNICA
C/ Alfonso Gómez,42 Nave 1-1-2
28037 Madrid



GRAFISMO PARA VIDEOJUEGOS

Preparados, Listos, ¡YA!
Autor: Antonio Marchal

Nivel: Básico

A partir del presente número, esta sección va a dar un giro de 360° para dedicarse a los videojuegos para grafistas, en lugar del grafismo para videojuegos, ya que vamos a comenzar un curso de programación en DIV, en el que, como práctica, se realizará un videojuego.

Estas lecciones pretenden ir dedicadas a todos aquellos que tengan unos conocimientos mínimos en programación, así como para aquellos otros que hayan leído el libro del DIV y les haya parecido un lenguaje de programación muy complicado. Por eso empezaremos por el principio. Antes de comenzar, queremos reseñar un pequeño error que se nos pasó por alto: los pequeños listados del mes pasado no funcionaban por un error tipográfico. Muchos lo habrán solucionado, mientras que, para otros, el error estaba en que sobrara un "1" que aparecía detrás de "variable_de_sonido" y delante del símbolo =. Si borramos dicho número, el programa funcionará correctamente, con lo que el problema se habrá solucionado. Pedimos mil perdones por lamentable error; el editor ya está sufriéndolo en el potro de torturas donde estará unas pocas semanas.

FILOSOFÍA DIV

Antes de entrar en materia, vamos a comentar la filosofía de funcionamiento que usa DIV Games Studio a la hora de trabajar. Como se explicó el mes pasado, un juego se compone de varias partes. Por un lado está el programa, también llamado código fuente o listado; estos tipos de archivos tendrán extensión, .PRG, están en formato texto y constituye el corazón de nuestro juego. En estos listados escribiremos las distintas instrucciones y haremos los distintos cálculos que darán vida a nuestra creación. Por otro, tenemos los gráficos, que se pueden almacenar de dos maneras: solos en formato con extensión .MAP, o agrupados en ficheros .FPG.

Un programa es un listado de instrucciones que se constituye como lo más importante de nuestro juego, ya que será el que se ocupe de insertar y quitar gráficos, moverlos y de todo tipo de interactividad con el usuario. Para funcionar, DIV Games Studio usa lo que ha denominado proceso. Estos procesos ya los veremos más detenidamente qué son y para qué se usan pero, primeramente, dejaremos claro un concepto como es el de las variables. Una variable es como un casillero en el cual podemos guardar un valor. A dicho casillero le asignaremos un nombre para, a lo largo del listado, referirnos a dicho casillero por el nombre. Dentro de DIV sólo se usa un tipo de variable, aunque existen derivaciones, que no serán explicadas hasta que sea necesario, que tiene unos valores máximos y mínimos; éstos son: -2147483648 y +2147483647, valores que no debemos sobrepasar, aunque ees complicado pues son valores muy altos, si bien hay que tener precaución. Por otro lado, existen tres espacios distintos para declarar las variables; dependiendo de si declaramos la variables en un sitio u otro éstas tendrán distinto funcionamiento. Según esta característica las variables pueden ser globales, locales o privadas. Cómo declarar una variable con cada característica lo veremos más adelante, cuando hablemos de los procesos. De las variables sólo veremos el uso en general, cómo declararla y usarla. Por ejemplo, para crear una variable que guarde el número de vidas, usaremos el nombre "*numerodevidas*", y asignaremos al inicio el valor 3, de la siguiente manera:

```
numerodevidas=3;
```

Desde ese momento, la variable *numerodevidas* valdrá 3 y si, por ejemplo, queremos multiplicar el número de vidas por dos, escribiremos:

```
numerodevidas = numerodevidas * 2;
```

Hay que reseñar el uso del símbolo punto y coma para señalizar el final de la línea; a partir de ahí, lo que se escriba será una nueva línea, independiente de la anterior. También podemos usar la mayoría de operaciones matemáticas con el uso de símbolos; en el ejemplo se aprecia cómo se ha utilizado el símbolo para multiplicar, pero hay otros como la suma, la resta, la divi-

sion, etc.. Más adelante se hará un estudio más detenido del uso de la variables, pero vamos a pasar a hablar de los procesos.

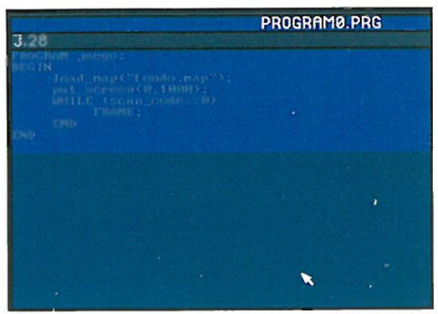
Un proceso puede cumplir cualquier función, y podría definirse como un conjunto de instrucciones que cumplen un cometido concreto, aunque como DIV está orientado a los videojuegos, los procesos también lo estarán, por lo que luego veremos características de estos procesos que estarán orientadas a este mundo. Para declarar un proceso deberemos usar la palabra *PROCESS* para, seguidamente, ponerle un nombre, que será con el que denominaremos desde ese momento a ese proceso y, por último, unos paréntesis. Del uso de los paréntesis ya se hablará, pero podemos adelantar que son obligatorios.

En cuanto al nombre, al elegir uno debemos decantarnos por alguno que tenga significado. Por ejemplo, si el proceso se va a ocupar de la nave principal, un buen nombre sería *nave_principal*, o *naveprincipal* o, simplemente, *navepr*. Hay que reseñar que el nombre debe empezar por una letra y no tiene que tener espacios; de ahí viene el uso del símbolo de subrayado. Después de esta declaración pueden aparecer declaraciones de variables y después de esto vienen las palabras BEGIN; luego vendrán las distintas instrucciones que elijamos para que realicen la opciones deseadas y, por último, la palabra END. Todo esto en conjunto quedaría así....

```
PROCESS nombre()
// Aquí puede ir declaraciones de variables
BEGIN
// Aquí está el código de instrucciones del proceso
END
```

Esta declaración es una versión simplificada de la real, pero que usaremos por ahora para explicar las distintas partes de un programa. Los paréntesis en este caso son obligatorios siempre, y su función no es otra que el paso de parámetros. Un parámetro es una variable que recibimos desde la llamada al proceso, justamente dentro de los paréntesis..

Existe un proceso especial, que podríamos denominar proceso inicial, que no comienza con la palabra *PROCESS*, sino con *PROGRAM*, y es el que designa el programa principal. En este caso no se hace uso de los paréntesis, debido a su carácter "especial". Aparte, este proceso, que a partir de ahora dejaremos de llamarlo así para denominarlo programa principal, puede tener más declaraciones de variables que un proceso normal, entre la declaración del mismo y la palabra *BEGIN*. La mayoría de estas declaraciones de variables, aparte de otras declaraciones avanzadas, no serán iguales que las de un proceso, y ya hablaremos más adelante del cometido de las mimas. Asimismo, existen otra serie de procesos predefinidos por DIV Games Studios; a este tipo de procesos los llamaremos a partir de ahora funciones. Por último,



Un simple programa para mostrar una imagen por pantalla.

señalar que todos los procesos, menos las funciones, tienen una serie de variables predefinidas que guardan valores importantes para el sistema.

Por otro lado, existen una serie de palabras reservas. Estas palabras serán las que controlen el camino que seguirá el programa, es decir, si ejecuta o no una parte de código, si lo tiene que repetir, etc... Alguna de estas palabras no son otras que *BEGIN, END, PROCESS, PROGRAM*, mientras que otras la iremos viendo poco a poco. Pero una de las palabras más importantes de todas ellas es *FRAME*. El cometido de esta palabra es la de volcar el contenido del proceso a pantalla, es decir, imprime los gráficos del proceso para que el usuario los vea. Esta palabra no es necesaria, pero esto sólo ocurre en algunos casos particulares.

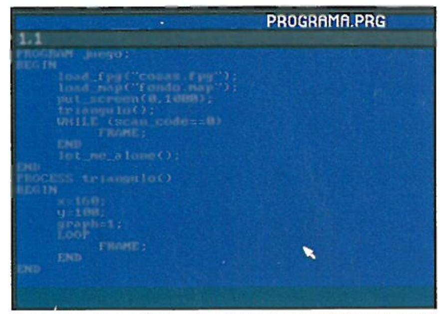
PROGRAMANDO QUE ES GERUNDIO

Después de haber visto las distintas partes de un programa, vamos a empezar con un ejemplo para comprobar cómo se usan cada una de estas partes. Lo primero es elegir una idea para nuestro juego; para el ejemplo usaremos gráficos, que serán figuras, pero que representarán naves, enemigos, coches o lo que la imaginación del lector quiera. La primera palabra que pondremos en nuestro proceso es *PROGRAM,(BEGIN ... END)* para indicar el nombre del programa, por lo que escribiremos en una nueva ventana de texto.

```
PROGRAM juego;
BEGIN

END
```

Luego introduciremos el código de nuestro programa, entre el *BEGIN* y el *END*, pero antes tenemos que crear algún trabajo gráfico. Primero realizaremos un fondo para nuestro juego de 320 por 200 puntos. Este fondo puede ser de montañas, estrellas o lo que se quiera. Una vez acabado debemos guardarlo en el disco duro para, después, usarlo en nuestro programa. El nombre que usemos al guardarlo debemos recordarlo para escribirlo luego en el programa. Para el ejemplo hicimos un fondo de estrellas, y lo guardamos con el nombre *fondo.MAP*; otra posibilidad hubiera sido haber optado por meter este gráfico dentro de otro de un fichero .FPG, pero eso ya lo haremos con el resto de grá-



Ejemplo de cómo se pueden programar los procesos en DIV.

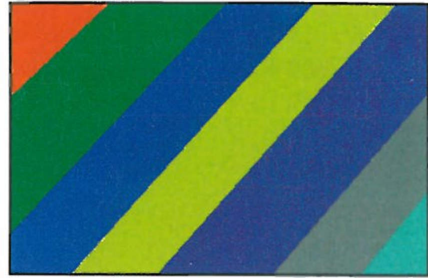
ficos, aunque siempre se puede optar por esta opción.

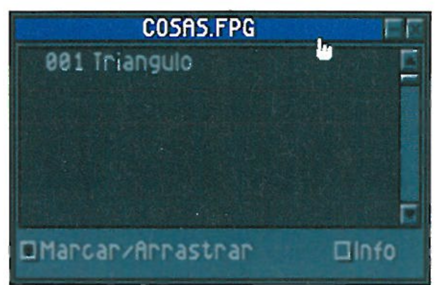
Ahora seguiremos escribiendo código, introduciendo nuevas líneas entre el *BEGIN* y el *END*; con estas líneas cargaremos la pantalla, la mostraremos en pantalla, y esperaremos a que se pulse una tecla. El código completo quedaría así:

```
PROGRAM juego;
BEGIN
load_map("fondo.map");
put_screen(0,1000);
WHILE (scan_code==0)
FRAME;
END
END
```

Pasaremos a comentar sólo el código nuevo, que comienza con la línea *load_map("fondo".map")*; esta línea está compuesta por una llamada a una función de las que dispone DIV. Dicha función no tiene otro cometido que el de cargar un gráfico en memoria. Para que el sistema sepa qué gráfico cargar, debemos pasar como parámetro el nombre del gráfico en el disco.

Por defecto, DIV Games Studios buscará el gráfico en varios lugares dentro del árbol del disco duro, aunque podemos especificar una ruta concreta; lo más recomendable es ir guardando cada cosa en el directorio del mismo nombre que su extensión. Esta función devuelve un valor, que podríamos guardar en una variable, pero, en este caso, para no complicar el listado no guardaremos dicho valor en ningún sitio. En la siguiente línea se usa otra función predeterminada de DIV; esta no es otra que *put_screen()*, a la cual se le deben pasar dos parámetros. El primero es el código del fichero gráfico, que es el fichero donde está el gráfico que vamos a usar. Pero como en nuestro caso hemos cargado directamente el gráfico en memoria, de manera autónoma, es decir, no pertenece a ningún fichero tipo .FPG sino que es un simple fichero .MAP. En este caso, tendremos que introducir como parámetro un 0, ya que el gráfico está cargado en memoria; si, por el contrario, estuviera compactado en el otro tipo de fichero, deberíamos indicar otro valor. Ahora pasemos a ver el otro parámetro. Con la anterior línea de código devolvía un valor que era el código del gráfico, puesto que este código de gráfico es el valor del segundo parámetro. Como se dijo anteriormente, no se almacenó dicho valor en ningún sitio; esto se hace en este caso


EL MAPA GRÁFICO UTILIZADO EN EL EJEMPLO.



NUESTRO FICHERO COSAS.FPG.

porque, al no tener más que un gráfico, su valor no es otro que 1000, que es valor del segundo parámetro. La función en sí tiene el cometido de poner un gráfico como fondo de pantalla. Esta función, como todas las gráficas, no tiene efecto hasta que encuentre la palabra *FRAME*, momento en el cual se efectuará la operación indicada.

La siguiente línea de código es algo más complicada pues, en principio, no es una función sino una palabra reservada, además está ligada a otra palabra para formar un conjunto que controle parte del código. Esta otra palabra es *END*, que si se observa en el listado, tiene la misma tabulación que la palabra *WHILE*. El código que está incluido entre estas dos palabras será dependiente de ellas, ejecutándose o no, dependiendo de las circunstancias que señalen dichas palabras. En el ejemplo hemos usado la palabra *WHILE*, que en inglés significa mientras, y sirve para hacer bucles, es decir, permite que partes de código se repitan. Cuando el programa llega a esta línea, comprueba lo que haya entre los paréntesis que, normalmente, es una línea que expresa una condición. Y mientras se cumpla dicha condición, el código que haya entre las palabras *WHILE* y *END* se repetirá, pudiéndose dar el caso de que nunca saliera, por ejemplo poniendo una condición que sea totalmente imposible. Por si no ha quedado claro el término de condición, se trata de una parte de código en el cual se comparan variables con otras variables y datos. En el ejemplo, lo que hacemos es coger una de estas variables, que está predeterminada, es decir, que ya ha sido creada por el sistema y puede contener un valor, cuyo nombre es *scan_code*, y cuyo cometido es guardar el código de la última tecla pulsada. Si este código es cero, es que no hay ninguna tecla pulsada. Sabiendo ya todo esto, la línea donde está el *WHILE* se podría traducir a un lenguaje más humano, de este modo, *"Mientras no se pulse ninguna tecla repite el siguiente código"*. El único código que se encuentra es otra palabra reservada, *FRAME*, que sirve para volcar el gráfico del proceso en pantalla. Esta palabra es necesaria en todas las funciones y procesos que tengan gráficos, o para aquellos que se repitan de manera cíclica, es decir, que no tengan un final predeterminado. Cuando el programa encuentra esta palabra, vuelca en pantalla todos los "trabajos" gráficos que se le hayan ordenado y proporcionará el control al siguiente proceso, así hasta que se ejecuten todos. En nuestro ejemplo continua, ya que sólo tenemos un "proceso" que no es tal, sino el programa principal y que, al encontrar la palabra *FRAME*, hará el único trabajo encomendado, que es el de poner el gráfico de fondo, que ordenamos que hiciera con la función *put_screen()*.

LA PANTALLA DE FONDO.



Un proceso puede cumplir cualquier función o un conjunto de instrucciones que cumplen un cometido concreto

Las líneas de código que existen a partir del *FRAME* es la palabra *END* y tiene el cometido de indicar el final de una parte de código en particular. Es recomendable tabular la palabra *END* a la otra palabra a la que estén ligados para, de esa forma, reconocer fácilmente donde empieza y acaba cada parte de código. Con esto finaliza el comentario del listado, por ahora, pues sólo queda pulsar la tecla de F10 para que el programa se compile y ejecute, además de ver los resultados. Si todo ha ido bien tendremos nuestro gráfico de fondo en pantalla, y en el momento que pulsemos una tecla, el programa saldrá, porque se ha cumplido la condición del *WHILE* y devolverá el control al sistema.

Ahora incluiremos un proceso propio, muy básico, para ver cómo funcionan. Como pasaba en el caso anterior, debemos crear primero el trabajo gráfico. Esta vez, en vez de ficheros gráficos sencillos, los de extensión -MAP, optaremos por los ficheros que son compilaciones de gráficos y cuya extensión es .FPG. Para crear este tipo de ficheros usaremos el menú ficheros, y la opción "Nuevo...", lo que nos pedirá un nombre para nuestro fichero; en nuestro ejemplo será *cosas.fpg*, y creará un fichero nuevo de este tipo. En este fichero incluiremos el resto de gráficos, e incluso podíamos haber incluido el del ejemplo anterior, cuestión que no se ha realizado para ejemplificar el uso de fichero gráficos de extensión .MAP, ya que, inicialmente, todos los gráficos son de este último tipo siendo, por el contrario, los otros ficheros de extensión .FPG, única compilación de los primeros. Para incluir un mapa gráfico dentro un fichero, se debe arrastrar dentro de este último; a la hora de arrastrar un gráfico hay que pulsar el botón izquierdo del ratón encima del gráfico, dentro del entorno, y sin soltar el botón del ratón mover el mapa gráfico hasta el fichero donde lo queremos guardar. Desde este momento, y para distinguir cada uno de los tipos gráficos, a los que vayan solos y tengan una extensión .MAP, los llamaremos mapas gráficos, y los otros, los que son compilación de estos recibirán el nombre de ficheros de gráficos. Para el ejemplo debemos crear otro gráfico que incluiremos dentro de nuestro fichero; en nuestro caso dibujaremos un simple triángulo relleno, al que asignaremos el código 1, dentro del fichero *cosas.FPG*, que nos servirá de ejemplo. Una vez tengamos este triángulo en el fichero, continuaremos con la tarea de programación..

INCLUYENDO FICHEROS Y PROCESOS

Lo primero que haremos en nuestro programa será cargar el fichero de gráficos que hemos creado; para ello usaremos una función muy parecida a la utilizamos con los mapas gráficos. Esta función es *load_fpg ("cosas.fpg")*, que nos devolverá un código que representará al fichero de gráficos cargado. En nuestro ejemplo, como ocurría en el caso anterior y para simplificar el listado, no asignaremos este código a ninguna variable, ya que sólo tenemos un fichero de gráficos cargado, y su código será el 1. Una vez incluida esta línea en el listado, el lugar donde la incluyamos puede ser, por ejemplo, delante de la línea del *load_map()*, pasaremos a la creación del proceso. Para ello, nos iremos al final del listado y escribiremos el siguiente código:

```
PROCESS triangulo()
BEGIN
x=160;
y=100;
graph=1;
LOOP
FRAME;
END
END
```

Pero vamos a comentar el código; las primeras líneas ya se comentaron cuando se habló de la declaración de procesos. Así, llegamos a la primera línea de código, de la que no hemos comentado nada, y es "*x=160;*". La primera deducción que hacemos es que con esta línea de código asignamos a "*x*" el valor 160. Pero, ¿qué es "*x*"? Pues es una variable predefinida del sistema que guarda la coordenada *x*, que es

la horizontal del proceso. Inicialmente vale 0, pero si modificamos este valor, también modificaremos la posición del proceso. Con la siguiente línea de código pasa lo mismo pero, en este caso, con la variable predefinida "*y*", que guarda la coordenada vertical del proceso. Por eso, pasaremos a la siguiente línea, la de "*graph=1;*", que es otra asignación a una variable predefinida, cuyo nombre es "*graph*", y que guarda el código del gráfico del proceso. Este código es el orden del mapa gráfico dentro de la memoria o dentro del fichero de gráficos cargado, del tipo FPG. Existe otra variable predefinida, muy ligada a ésta, que no usaremos por ahora y cuyo nombre es "*file*", que guarda el código de fichero gráfico del proceso. Los códigos que asignaremos a esta última variable serán parecidos a los usados como primer parámetro de la función *put_screen*, y los usados con la variable "*graph*" serán equiparables a los del segundo parámetro.

Una vez visto esto, comentaremos lo que queda de código del proceso ya que está todo muy ligado. Lo único reseñable es la palabra reservada *LOOP*, la cual tiene ligada un *END* para designar una parte de código, como ocurría con *WHILE*. En el caso de *LOOP*, que traducido del inglés significa algo así como "vuelta", lo que se consigue es un bucle infinito, es decir, que no tiene final. El único código que se ha incluido dentro de este bucle infinito es la palabra reservada *FRAME*, que hará que el gráfico del proceso aparezca en pantalla, como en el caso del programa principal.

Todas estas líneas serán el código del proceso, al que deberemos llamar desde el programa principal. Esta llamada la conseguiremos incluyendo la línea:

triángulo();

justo delante de la línea del *WHILE* con lo que, cuando el programa principal llegue a esta línea, que llama al proceso, ejecutará el código del proceso hasta que encuentre un *FRAME*, o el proceso acabe, momento en el que el sistema continuará con la lectura del código del programa principal, hasta que encuentre un *FRAME*, o se acabe el programa. Si no se ha entendido muy bien lo de los saltos en la lectura del programa, no hay que preocuparse, ya que se realizará un seguimiento más descriptivo en siguientes lecciones del curso. Lo último que nos queda por resolver es el problema de haber dejado al proceso que hemos creado en un bucle infinito. Para resolver esto, utilizaremos una función llamada "*let_me_alone()*", que viene a significar algo así como "dejarme solo", y cuyo cometido es el de eliminar todos los procesos que se estén ejecutando, menos el que posea la instrucción que, en nuestro caso, será el programa principal. Esta línea de código, con la función que queremos usar, la debemos incluir detrás del *END*, que está ligado al *WHILE* del programa principal, que es el punto del código donde se ha pulsado una tecla y, por lo tanto, se quiere salir del programa; con esta función eliminaremos todo lo que esté pendiente de ejecución. Para clarificar todo, vamos a ver cómo quedaría el programa completo:

```
PROGRAM juego;
BEGIN
load_fpg("cosas.fpg");
load_map("fondo.map");
put_screen(0,1000);
WHILE (scan_code==0)
FRAME;
END
let_me_alone();
END
PROCESS triangulo()
BEGIN
x=160;
y=100;
graph=1;
LOOP
FRAME;
END
END
```

Cuando pulsemos de nuevo F10, ejecutaremos el programa y veremos que aparece un triángulo, que es el gráfico del proceso, encima del fondo que pusimos con la función *put_screen()*. Igual que pasaba en el caso anterior, en el momento que pulsemos una tecla, el programa acabará y devolverá el control al sistema.

CONCLUSIÓN

Por último, vamos a hacer un repaso de los diferentes conceptos que hemos visto en el artículo. Programa principal, proceso, función predefinida, variables, variables predefinidas, palabras reservadas, mapa gráfico, fichero de gráficos, parámetro, bucle infinito o bucle condicional son algunos de los conceptos que debemos recordar, aunque en sucesivos textos se realice un estudio más exhaustivo de los mismos. Si surgiera alguna duda podéis dirigiros al autor vía e-mail: *tizo@100mbps.es.*

En el próximo artículo empezaremos a mover nuestro triángulo. Tan solo decir que para los que no quieran escribir el código de los ejemplos del artículo, existe en el CD una copia del mismo, una lista para usar, pero lo único que hay que hacer es copiar los distintos ficheros al directorio del mismo nombre que la extensión de dichos ficheros.

UN POCO DE ARGOT

Funciones predefinidas: Son una serie de funciones con distintos cometidos; podremos ver más información sobre ellas en la ayuda o en el libro de DIV Games Studio. En el artículo hemos visto algunas como éstas:

```
put_screen(); // Pone una pantalla de fondo
load_map();  // Carga un mapa gráfico
load_fpg();  // Carga un fichero de gráficos
```

Palabras reservadas: Este tipo de palabra, predefinidas por el lenguaje, nos sirven para indicar qué caminos deben tomar los programas, y para diferenciar las distintas partes del código; algunas de estas palabras tienen varios cometidos y son usadas conjuntamente con otras.

```
PROGRAM // Indica el comienzo del programa
PROCESS // Indica el comienzo de un proceso
BEGIN   // Indica dónde comienza el código de un proceso o programa
END     // Indica el final de una parte del código
WHILE   // Permite que se repita una parte del código condicionalmente
LOOP    // Permite que se repita una parte del código indefinidamente
FRAME   // Hace un vuelco del proceso a pantalla
```

Variables predefinidas: Son una serie de variables que guardan valores y que están predefinidas por el sistema; como en el caso de las funciones, existe bastante documentación al respecto, pero enumeremos las usadas en el artículo:

```
scan_code // Guarda el código de la última tecla
x     // Guarda la coordenada horizontal de un proceso
y     // Guarda la coordenada vertical de un proceso
graph // Guarda el código del gráfico del proceso
```

Mapa Gráfico: Son aquellos que se pueden guardar en formato .MAP, pudiendo ser importados y exportados a los formatos .BMP y .PCX. Si cargamos el gráfico en memoria empezará su código a partir de 1000.

Fichero de gráficos: Es un compendio de mapas gráficos que se almacenan de manera autónoma. En estos ficheros cada mapa gráfico tendrá un orden, que será el valor que usemos a la hora de hacer reseña a los mismos.



WORKSHOP PROGRAMACIÓN

PC

Optimizando OpenGL

Autor: Roberto López

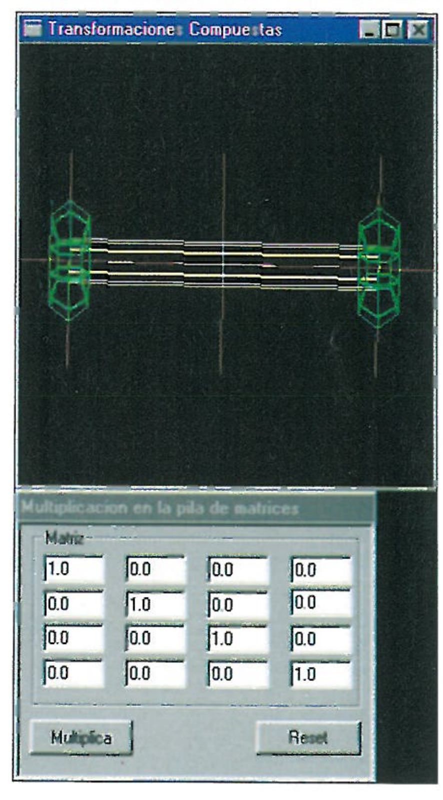
Nivel: Avanzado

En la entrega anterior se mostró al lector la forma en que opera la pila de matrices de transformación en OpenGL y los comandos mediante los cuales es posible manipular las diferentes pilas existentes. Al estudiar estos comandos se insistió en su correcta utilización, pues OpenGL opera con ellos de manera muy eficiente.

Sin embargo, en caso de escenas complejas donde se manipulen muchos objetos con un número elevado de vértices, no basta con manejar la pila de matrices de acuerdo a las mejores recomendaciones, si se pretende obtener animaciones fluidas y garantizar una elevada interactividad en tiempo real. Para tales situaciones OpenGL dispone de otras herramientas, y en particular de las llamadas *Display List* que serán objeto de análisis en la presente entrega.

El *Display List* no es más que un medio que tiene OpenGL de almacenar una secuencia de comandos para su posterior ejecución. Cuando se invoca un *Display List*, los comandos que ésta ha encapsulado se ejecutan en el mismo orden en que fueron introducidos. El empleo de este tipo de instrucción resulta muy efectivo cuando se redibuja la misma geometría múltiples veces y cuando se tiene un conjunto de cambios de estado que es necesario aplicar reiteradamente.

PROGRAMA DE EJEMPLO INCLUIDO EN EL CD-ROM.



Un *Display List* contiene únicamente comandos de OpenGL

El programa que acompaña esta entrega muestra el uso del *Display List*. En los recuadros 1 y 2 se presentan las secuencias de instrucciones para dibujar un eje cilíndrico y dos ruedas en sus extremos. En el segundo recuadro se realizan las mismas operaciones que en el primero, pero utilizando *Display Lists*. Salta a la vista la drástica reducción del número de instrucciones que tiene que ejecutar OpenGL en el segundo caso.

CREACIÓN DEL *DISPLAY LIST*

Si el lector abre el programa fuente podrá observar que en la rutina de dibujado aparece la secuencia normal de instrucciones limitada al inicio y al final por las instrucciones **glNewList( )** y **glEndList( )**, que definen el *Display List*. El comando **glNewList** utiliza dos argumentos, un índice entero y un parámetro que puede tomar los valores GL_COMPILE y GL_COMPILE_AND_EXECUTE. El primero indica solo la creación del *Display List*, y el segundo, la creación y ejecución inmediata del mismo.

Un *Display List* contiene sólo comandos de OpenGL. En el programa de ejemplo se encapsulan comandos de definición de vértices y color de los ejes, y de operaciones de traslación y rotación, así como el propio dibujado de las piezas. Los parámetros de cada llamada de comando se evalúan y sus valores se copian en el *Display List* cuando es creado. Toda la trigonometría necesaria para crear el cilindro y los toroides se realiza una sola vez, lo que incrementa la rapidez de ejecución. Sin embargo, una vez que un comando ha sido introducido en el *Display List* no es posible eliminarlo ni añadir nuevos comandos a la lista una vez que ha sido creada. Se puede destruir un *Display List* y construir uno nuevo, pero en ningún caso se puede editar.

Aunque no se garantiza que una implementación dada de OpenGL optimice el *Display List* en cualquier caso, su ejecución nunca será más lenta que la ejecución de los comandos que la componen. Los casos más susceptibles de ser optimizados mediante el *Display List* se relacionan continuación:

- Operaciones de matrices.
- *Raster* de mapas de bits y de imágenes.
- Luces, materiales y modelos de iluminación.
- Texturas.

Existen algunos comandos que no pueden ser almacenados en un *Display List*. La principal razón es la concepción cliente-servidor en que funciona OpenGL en red. Después de ser creado, el *Display List* queda residente en el servidor, de manera que este no puede depender del cliente para la ejecución del *Display List*. Así, ningún comando que devuelva valores, como los del tipo glGet*( ) y glIs*( ), puede ser almacenado en el *Display List*. Tampoco los comandos que cambien el estado del cliente como glPixelStore(), glSelectBuffer() y los comandos que definen los llamados *Vertex Arrays* pueden ser almacenados en un *Display List*.

EJECUCIÓN DEL *DISPLAY LIST*

Una vez creado el *Display List* se puede ejecutar llamando a la función **glCallList()**, que utiliza como argumento el mismo índice entero con que fue creado. La llamada a glCallList() se puede realizar desde cualquier parte del programa, mientras esté activo el contexto de OpenGL en que se creó el *Display List*. Un *Display List* puede ser creado en una rutina y ser ejecutado en otra, ya que el índice lo define unívocamente.

Existen algunos comandos que no pueden ser almacenados en un *Display List*

Un *Display List* también puede ejecutar otros *Display Lists*, formando lo que se llaman *Display Lists* jerárquicos, como se muestra a continuación:

```
GlNewList(indice1, GL_COMPILE);
   GlCallList(indice2);
   GlCallList(indice3);
   GlRotate(1.0, 0.0, 0.0, 30.0);
   GlCallList(indice4);
glEndList();
```

Para evitar llamadas, recursivas infinitas existe un límite al numero de niveles de llamadas recursivas. Para determinar ese límite se debe realizar la siguiente llamada:

GlGetIntegerv(GL_MAX_LIST_NESTING, Glint *data);

OpenGL permite también la ejecución sucesiva de un número de *Display Lists*. Este mecanismo exige que inicialmente se coloquen en un arreglo los índices de los *Display Lists* y se llame posteriormente al comando **glCallLists()** para la ejecución. Este comando tiene la siguiente sintaxis:

glCallLists(Glsizei *n*, Glenum *type*, const Glvoid **list*);

El argumento *n* indica el número de *Display List* que se ejecutarán, *type* indica el tipo de dato de los valores en la lista (GL_BYTE, GL_UNSIGNED_BYTE, GL_SHORT, ...).

De la misma forma que se pueden encapsular un conjunto de instrucciones mediante *Display List,* éste comando se utiliza también para definir mediante un número comandos, un estado dado, y cambiar a otro estado definido por otro conjunto de comandos. Cuando se quiere cambiar de un estado a otro resulta rentable el empleo de un *Display List* donde se han almacenado previamente las instrucciones que definen el nuevo estado.

OpenGL permite la ejecución sucesiva de varios *Display List*

Como se ha podido apreciar, el *Display List* es una técnica que bien empleada puede reportar un beneficio neto de incremento de la eficiencia del programa. Se estimula al lector a comprobar por sí mismo sus beneficios, incluyendo su uso en los nuevos programas que escriba. En nuevas entregas seguiremos analizando otras formas de optimizar la ejecución de OpenGL. Como es usual, junto con el programa fuente, el lector puede encontrar el ejecutable, así como las librerías necesarias para poder compilarlo en vuestro ordenador.

SECUENCIA NORMAL DE INSTRUCCIONES DE DIBUJADO

```
void CALLBACK display(void)
{
glClear(GL_COLOR_BUFFER_BIT);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
//se multiplica por "matriz", la matriz que edita el usuario
//que por defecto es la identidad
glMultMatrixf(matriz);
gluLookAt(radio[0], radio[1], radio[2],
      0.0, 0.0, 0.0,0.0, 0.0,1.0 );
glTranslatef (0.0, 0.5, 0.0);
glColor3f (1.0, 1.0, 1.0);
/7dibuja el cilindro
auxWireCylinder(0.2, 3.0);
//Guarda la matriz de transformacion
glPushMatrix();
glRotatef (90.0, 1.0, 0.0, 0.0);
glTranslatef (0.0, 0.0, -1.0);
glRotatef ((GLfloat) rot, 0.0, 0.0, 1.0);
glColor3f (0.0, 1.0, 0.0);
//dibuja la primera rueda
auxWireTorus(0.2, 0.35);
//Recobra la matriz de transformacion guardada
glPopMatrix();
glRotatef (90.0, 1.0, 0.0, 0.0);
glTranslatef (0.0, 0.0, 2.0);
glRotatef ((GLfloat) rot, 0.0, 0.0, 1.0);
glColor3f (0.0, 1.0, 0.0);
// dibuja la segunda rueda
auxWireTorus(0.2, 0.35);
glFlush();
}
```

Recuadro 2. Secuencia de instrucciones utilizando *display list*

```
if(glIsList(1)==GL_TRUE )
{
//Lista 1
glCallList(1);
glPushMatrix();
glRotatef (90.0, 1.0, 0.0, 0.0);
glTranslatef (0.0, 0.0, -1.0);
glRotatef ((GLfloat) rot, 0.0, 0.0, 1.0);
//Lista 2
glCallList(2);
//Recobra la matriz de transformacion
glPopMatrix();
glRotatef (90.0, 1.0, 0.0, 0.0);
glTranslatef (0.0, 0.0, 2.0);
glRotatef ((GLfloat) rot, 0.0, 0.0, 1.0);
//Lista 3
glCallList(3);
}
```

```
if(glIsList(1)==GL_TRUE )
    {
    //Lista 1
    glCallList(1);
    glPushMatrix();
    glRotatef (90.0, 1.0, 0.0, 0.0);
    glTranslatef (0.0, 0.0, -1.0);
    glRotatef ((GLfloat) rot, 0.0, 0.0, 1.0);
    //Lista 2
    glCallList(2);
    //Recobra la matriz de transformacion
    glPopMatrix();
    glRotatef (90.0, 1.0, 0.0, 0.0);
    glTranslatef (0.0, 0.0, 2.0);
    glRotatef ((GLfloat) rot, 0.0, 0.0, 1.0);
    //Lista 3
    glCallList(3);
    }
```



LIGHT WAVE

Utilizando el entorno de trabajo
Autor: José María Ruíz

Nivel: Medio

Es extremadamente importante poder manejar todas las herramientas de manera apropiada y cómoda. El ajuste de las vistas, del zoom, controlar los desplazamientos de las vistas, la ocultación y reaparición de polígonos forman parte del control del entorno de trabajo.

Para ajustar una porción de la pantalla se debe pulsar sobre el botón *Magnify* del menú *Display,* a continuación se colocará el puntero del ratón sobre la zona que se desee ampliar o reducir y se pulsará el botón izquierdo del ratón, manteniéndolo pulsado; en este momento, si el ratón se desplaza hacia la derecha la vista se ampliará, si, por el contrario, el desplazamiento es hacia la izquierda la vista se reducirá tomando siempre como centro de la ampliación el lugar donde se ha pulsado el botón izquierdo del ratón por primera vez.

Si, por ejemplo, se desea ampliar una esfera que aparece pequeña en la parte superior de una de las vistas se pulsará con el ratón en el centro aproximado de la esfera y, a continuación, manteniendo pulsado el botón izquierdo del ratón se llevará hacia la derecha hasta que ocupe el tamaño que convenga dentro de la vista.

AJUSTE DEL ZOOM EN UNA DE LAS VISTAS

Si se desea ampliar una porción de la pantalla se pulsará sobre el botón *Zoom* del menú *Displa*; a continuación se pulsará con el botón izquierdo del ratón en el centro de la zona que se desee ampliar, después se trazará una ventana manteniendo pulsado el botón izquierdo del ratón; la zona de la ventana que aparezca señalada será la que finalmente se ampliará.

En la figura 1 se puede ver en la ventana del perfil la caja amarilla que se ha trazado para hacer *Zoom,* mientras que en la figura 2 se puede ver el resultado del *Zoom.*

AJUSTE DEL DESPLAZAMIENTO DE LAS VISTAS

Cuando se desee desplazar la vista sobre los objetos que existan, es decir, mover la posición de la cámara sin ampliar ni reducir la vista, se utilizará el botón *Pan* del menú *Displa*; a continuación, se pulsará con el botón izquierdo del ratón sobre la vista que se desee desplazar manteniendo pulsado el botón. Moviendo entonces el ratón en cualquier dirección, se apreciará cómo el objeto se desplaza en la misma dirección en la que se desplaza el ratón. El atajo de teclado es la tecla "Alt".

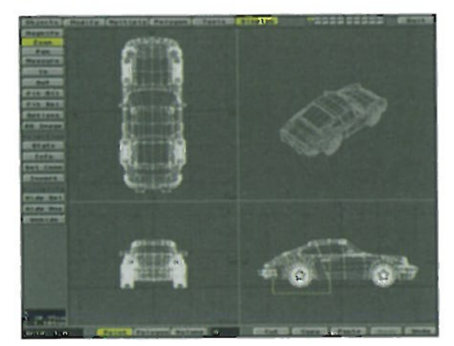


FIGURA 1. APLICACIÓN DE LA HERRAMIENTA *ZOOM*.

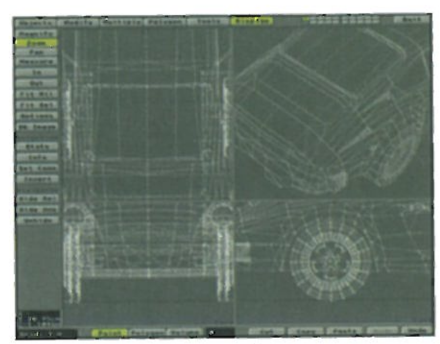


FIGURA 2. RESULTADO DEL *ZOOM*.

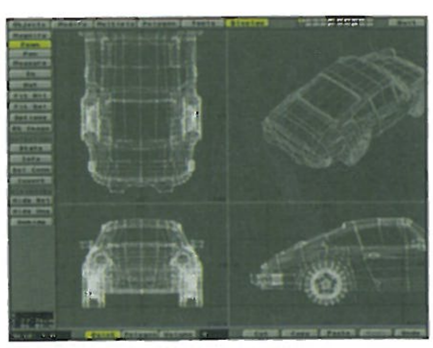


FIGURA 3. APLICACIÓN DE LA HERRAMIENTA *PAN*.

En la figura 3 se ha utilizado la herramienta *Pan* en la ventana *Left* (perfil) para poder ver la delantera del vehículo representado, y en la figura 4 se puede ver el resultado tras aplicar el desplazamiento lateral.

MEDICIÓN MANUAL

Para poder tomar medidas manualmente existe la herramienta *Measure* en el menú *Display*. Para efectuar una medición basta con llevar el ratón a la vista donde se encuentra lo que se desea medir, y se pulsa el botón izquierdo del ratón sobre el comienzo de lo que se desee medir, manteniéndolo pulsado hasta que se lleva al lugar donde termina la medición. Todavía sin soltar el ratón se puede ver el la parte inferior izquierda de la pantalla debajo de la inscripción *Length* el resultado de la medición.

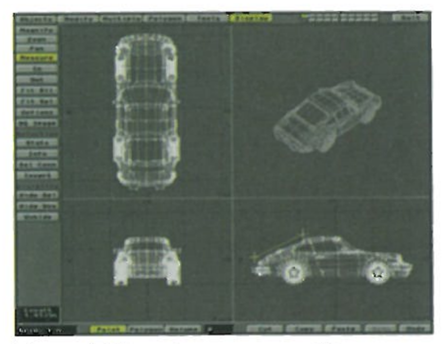


FIGURA 5. MEDICIÓN MANUAL CON *MEASURE*.

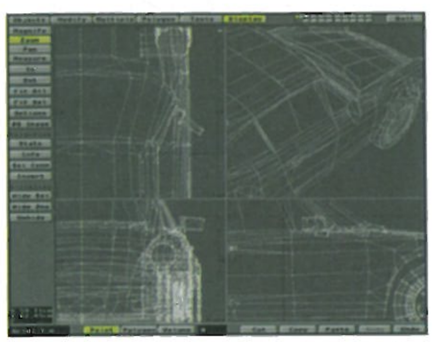


FIGURA 6. ANTES DE APLICAR AJUSTE AUTOMÁTICO A TODAS LAS VISTAS.

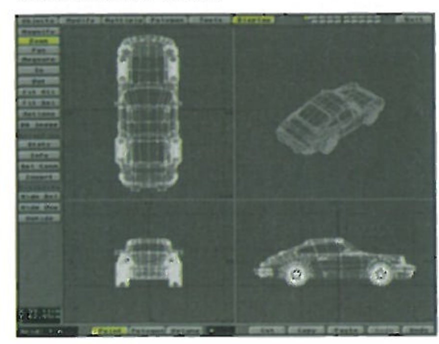


FIGURA 7. TRAS APLICAR EL AJUSTE AUTOMÁTICO.

En la figura 5 se puede ver una medición y el resultado en la esquina inferior izquierda de la pantalla.

AMPLIACIÓN DE LAS VISTAS

Cuando se deseen ampliar todas las vistas desde sus centros, la forma más rápida será pulsando el botón *In* del menú *Display*.

El tamaño de la ampliación en todas las vistas será de una unidad del cuadrado de la rejilla *(grid)* y el atajo de teclado es ".".

REDUCCIÓN DE LAS VISTAS

Por su parte, cuando se deseen reducir todas las vistas desde sus centros, la forma más rápida será pulsando el botón *Out* del menú *Display*.

El tamaño de la reducción en todas las vistas será de una unidad del cuadrado de la rejilla *(grid)* y el atajo de teclado es ",".

AJUSTE DEL OBJETO A TODAS LAS VISTAS

Cuando se desee ajustar el total del objeto a todas las vistas se pulsará sobre el botón *Fit All* del menú *Display*.

El atajo de teclado es "a".

En la figura 6 se puede ver la vista antes de ajustar el objeto a todas las vistas, mientras que en la figura 7 después de ajustarse.

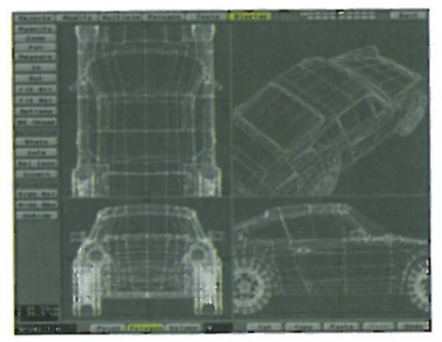


FIGURA 8. AJUSTE AUTOMÁTICO A UNA SOLA VISTA.

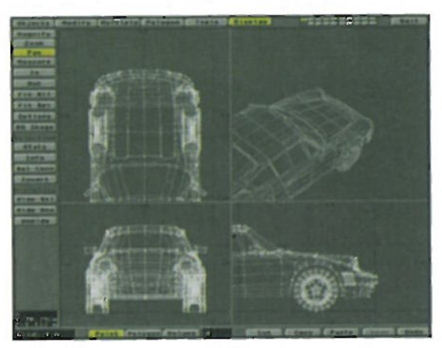


FIGURA 4. RESULTADO DE LA HERRAMIENTA *PAN*.

AJUSTE DEL OBJETO A UNA SOLA VISTA

Cuando se desee ajustar una vista del objeto se utilizará manteniendo el ratón sobre la vista que se desee ajustar y a continuación se pulsará el atajo de teclado "Ctrl+a".

En la figura 8 se puede ver cómo se ha ajustado el objeto a una sola vista, la vista del alzado *(face)*.

AJUSTE DE LO SELECCIONADO A LAS VISTAS

Cuando se desee ajustar lo seleccionado a todas las vistas se pulsará sobre el botón *Fit Sel* del menú *Display*.

El atajo de teclado es "Shift+a" (mayúsculas+a).

En la figura 9 se puede apreciar cómo el total de lo seleccionado se ha ajustado a todas las vistas tras aplicar la herramienta *Fit Sel*.

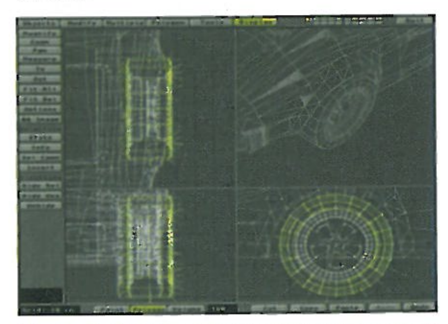


FIGURA 9. AJUSTE DE LO SELECCIONADO A TODAS LAS VISTAS.

OPCIONES DE VISUALIZACIÓN

Si se pulsa sobre el botón *Options* del menú *Display* se accede a las opciones de visualización, donde se pueden ajustar los siguientes parámetros:

- ***Orientation* (Tipo de Orientación).** En esta opción se puede cambiar la posición de las vistas, se puede elegir entre:

  - *Quad: Logo (XY)* Figura 10
  - *Quad: Map (XZ)* Figura 11
  - *Quad: Side (ZY)* Figura 12
  - *Quad: Vehicle* Figura 13
  - *Single: Face* (Alzado) Figura 14
  - *Single: Side* (Perfil) Figura 15
  - *Single: Map* (Planta) Figura 16
  - *Single: Preview* (Previsualización) Figura 17

El modelador de Lightwave permite colocar una imagen de cualquier vista

- ***Preview Type* (Tipo de previsualización).** Se puede elegir entre:

  - *None:* Desactivada
  - *Wire:* Modo "alambre"
  - *Frontface*: Sólo muestra los polígonos cuya cara normal sea visible en la perspectiva de la previsualización
  - *Solid:* Muestra una previsualización sólida
  - *Open GL Sketck*: Visualización en grises en modo alambre relleno
  - *Open GL Flat*: Visualización en color mostrando, además, la especularidad, el brillo y los polígonos como doble cara
  - *Open GL Smooth Shaded:* Visualización sólida con máximo suavizado y realismo
  - Las visualizaciones *Open GL* no son soportadas por el Amiga

- ***Visibility* (Ajuste de visibilidad):** En este parámetro se puede activar o desactivar la visibilidad de los siguientes elementos.

  - *Points*: Puntos
  - *Faces:* Polígonos
  - *Curves:* Curvas
  - *Normals:* Normales de los polígonos
  - *Grid:* Rejilla
  - *Backdrop:* Gráfico de fondo
  - *Patch Polygon*: Polígonos temporales
  - *Patch Surface*: Superficies temporales

- ***Screen* (Ajuste de resolución) (Sólo en Amiga):** Con esta opción se puede ajustar en los ordenadores Amiga el tipo de resolución gráfica, así como el tamaño de la misma.

- ***Interface* (Modo de color) (Sólo en Amiga):** Se puede elegir entre:

  - *Monochrome:* Monocromo
  - *Color:* Color

- ***Input Device* (Ajuste del digitalizador):** Se pueden seleccionar:

  - *Mouse:* Ratón
  - *Tablet:* Tableta digitalizadora

- ***Unit System* (Ajuste del sistema de medidas):** Se puede elegir entre:

  - *SI:* Sistema Internacional.
  - *Metric:* Sistema Métrico.
  - *English:* Sistema Inglés.

- ***Grid Units* (Unidades de la rejilla):** Se pueden elegir:

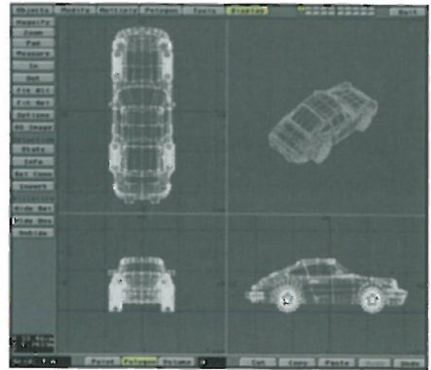
  - *1* : Enteros
  - *1 5* : Enteros y medios
  - *1 2.5 5*: Enteros, medios y cuartos
  - *1 2 5* : Enteros, dobles y medios
  - *1 2* : Enteros y dobles

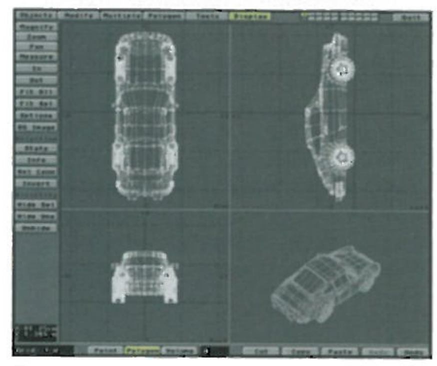
- **Grid Snap (Tamaño de la rejilla):** A elegir entre:

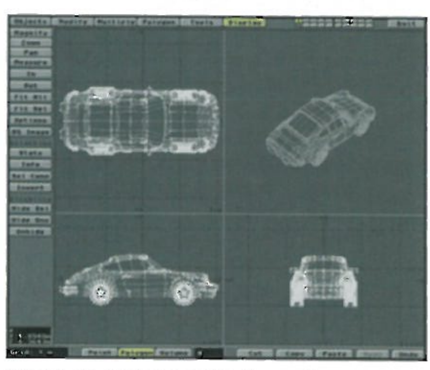
  - *None:* Sin rejilla
  - *Standard:* Se utiliza el valor seleccionado en *Grid Units*
  - *Fixed:* Se utilizará el valor introducido a continuación, admite magnitudes

AJUSTE DE UN GRÁFICO COMO FONDO EN UNA VISTA

El modelador del *Lightwave* permite colocar una imagen de cualquiera de las vistas, de esta forma se puede usar como referencia de modelado.


FIGURA 10. VISUALIZACIÓN *QUAD: LOGO.*


FIGURA 11. VISUALIZACIÓN *QUAD: MAP.*


FIGURA 12. VISUALIZACIÓN *QUAD: SIDE.*

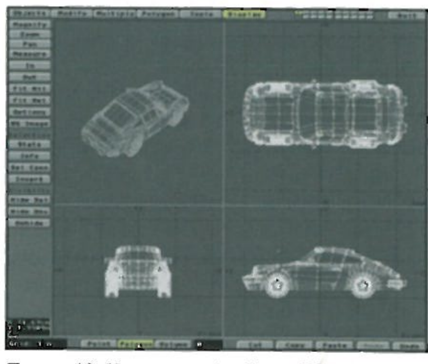
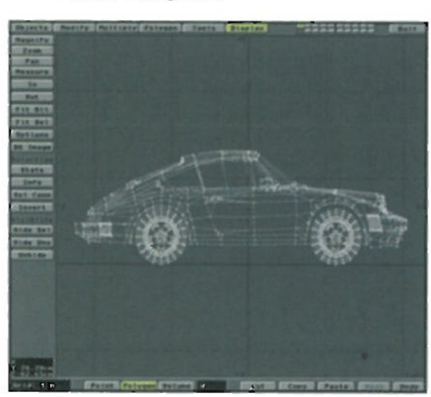
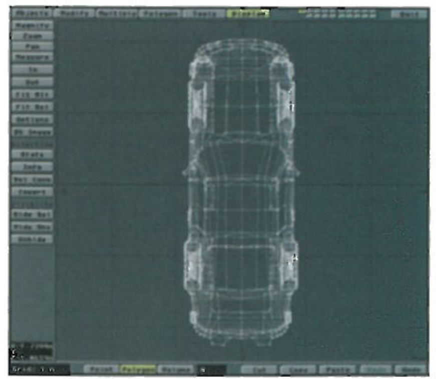

FIGURA 13. VISUALIZACIÓN *QUAD: VEHICLE.*


FIGURA 14. VISUALIZACIÓN *SINGLE: FACE.*


FIGURA 15. VISUALIZACIÓN *SINGLE: SIDE.*


FIGURA 16. VISUALIZACIÓN *SINGLE: MAP.*

PRÁCTICA Nº 17

Es recomendable practicar no sólo este ejercicio, sino todos los ejemplos comentados en este artículo. En esta práctica se va a realizar un asteroide:

1) Se empezará creando una esfera con muchas subdivisiones; en el ejemplo 32 *Sides* y 16 *Segments* tal y como se ve en la figura A.

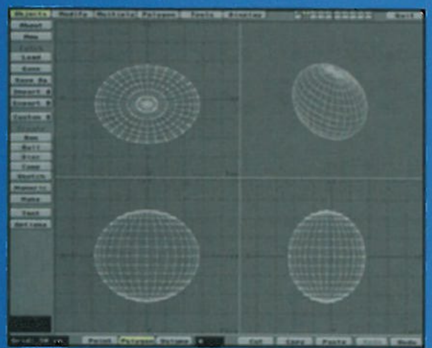


FIGURA A.

2) Con la herramienta *Magnet* se aplicaran sucesivas deformaciones, como se puede ver en la figura B.

3) Con las herramientas Pole 1 y Pole 2 se aplican nuevas deformaciones en todas las vistas tal y como se puede ver en la figura C.

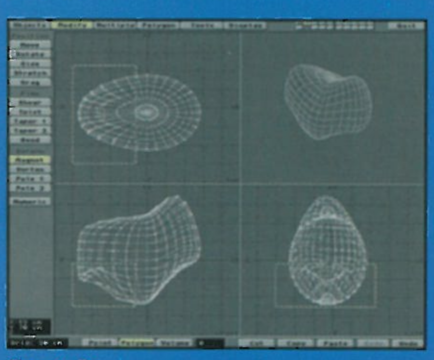


FIGURA B.

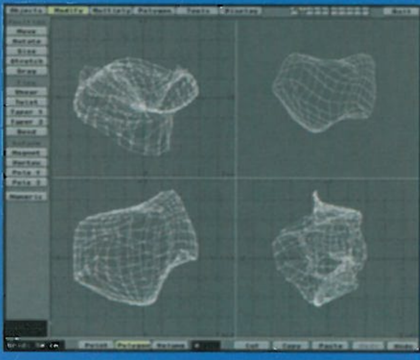


FIGURA C.

4) Después, se puede aplicar un Jitter Gaussian con un valor pequeño; el resultado será similar a la figura D.

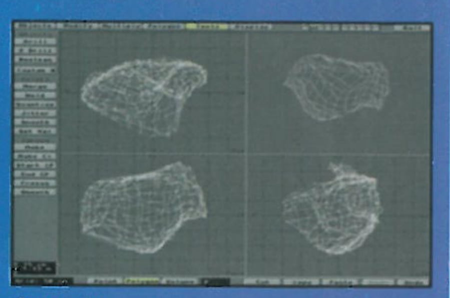


FIGURA D.

5) Finalmente, se pueden utilizar las herramientas *Shear, Taper1, Taper 2* y *Bend* para aplicarle las ultimas modificaciones. El resultado final será parecido a la figura E.

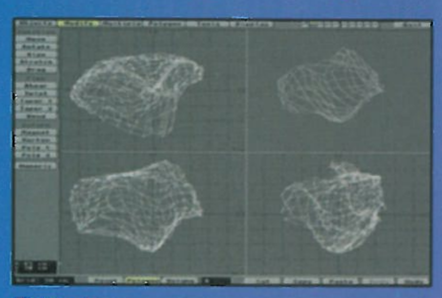


FIGURA E.

El Objeto terminado se encuentra en el CD-ROM dentro de un directorio llamado ARTIC\LIGTHWAVE y se llama ASTERIOD.LWO.

Si se pulsa sobre el botón *BG Image* del menú Display aparece una ventana de requerimiento de datos con las opciones siguientes:

Load: Para leer una nueva imagen.

Clear: Para borrar la imagen actual.

Axis: Elegimos el eje sobre la que se proyectará la imagen, cada eje representa una vista.

Center: Define un punto que será el centro de la imagen.

Size: En esta opción se podrá introducir el tamaño con el que se visualizará la imagen.

El botón *Options* del menú *Display* da paso a las opciones de visualización

Automatic Size: Si se pulsa este botón la imagen se adaptará automáticamente al tamaño del objeto actual.

Reset: Restaura los valores por defecto.

Light - Dark: Este desplazador sirve para ajustar la intensidad de iluminación con la que aparecerá dicha imagen.

Ok: Para aceptar los valores introducidos.

Cancel: Para anular los valores introducidos.

En la figura 18 se puede apreciar una imagen mostrada en una de las vistas.

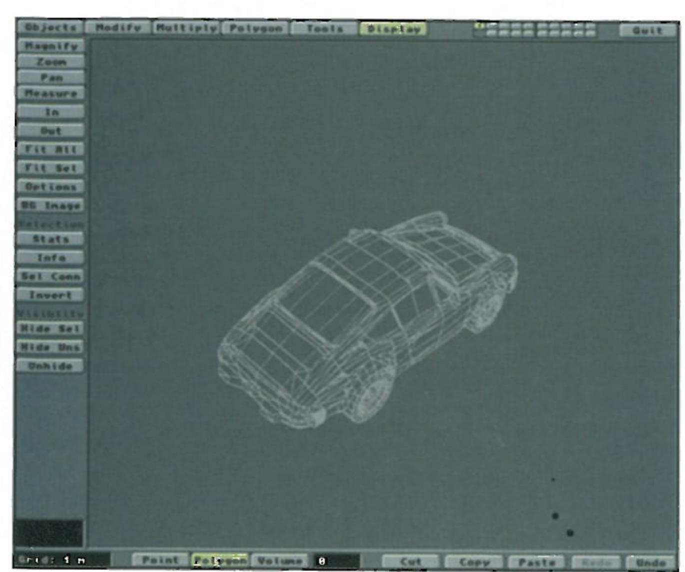
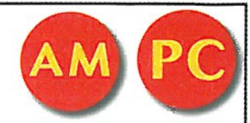


FIGURA 17. VISUALIZACIÓN *SINGLE: PREVIEW*.



FIGURA 18. AJUSTE DE UNA IMAGEN EN UNA VISTA.





REAL 3D

Operaciones con B-Splines
Autor: David Díaz

Nivel: Medio

En el presente capítulo se adentrará un poco más en el interminable mundo del modelado mediante curvas y superficies B-Spline, desarrollando nuevas técnicas y metodologías de modelado básico por medio del uso de objetos de apoyo adicional.

A su vez, se introducirá un concepto nuevo, el de la pila de datos que, en algunas ocasiones, puede salvar al usuario de tener que depender de su habilidad ante el ratón para determinar un buen o mal modelado.

Con ello, se pretende comenzar a integrar conocimientos de diferente índole de este maravilloso programa que, día a día, se transforma más y más en una herramienta para crear gráficos 3D imprescindible.

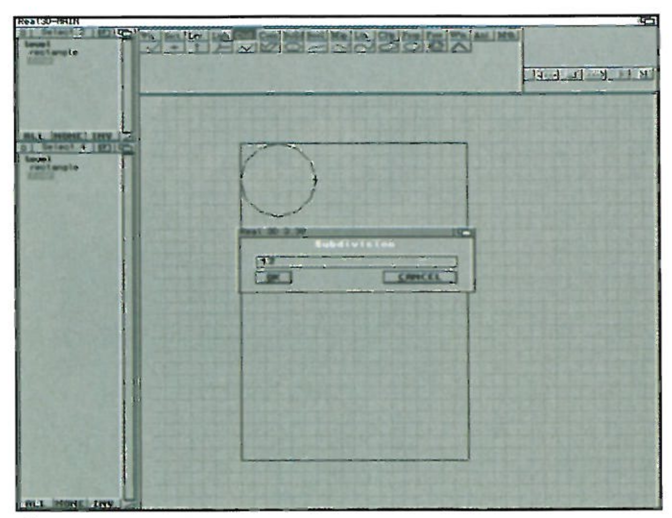
AHONDANDO EN LAS B-SPLINE

Bien es sabida la flexibilidad y el ingente número de posibilidades que ofrecen en el modelado las superficies y curvas B-Spline. También es sabido que son las entidades más maleables que conforman Real3D. No obstante, es posible el caso en el que, al enfrentarnos a un modelado determinado, el usuario decida pasar por alto ciertas apreciaciones y se sienta finalmente inseguro por no conseguir un resultado con la precisión y fiabilidad que el modelo original requiere.

Vamos a imaginar el modelado de la base de un teléfono. A priori, es una especie de "caja" con forma de cajetín de cerillas, pero con los bordes romos y en cuya zona superior hay dos espacios correspondientes para introducir los salientes del auricular y del altavoz. Para conseguir este modelo, convendría primero valorar si las propiedades B-Spline, que no son incluidas en las primitivas, van a ser necesitadas a posteriori. Es decir, si el objeto será rígido, y en el futuro no necesitará de deformaciones no lineales, es aconsejable el intentarlo mediante primitivas, con el fin de ahorrar recursos.

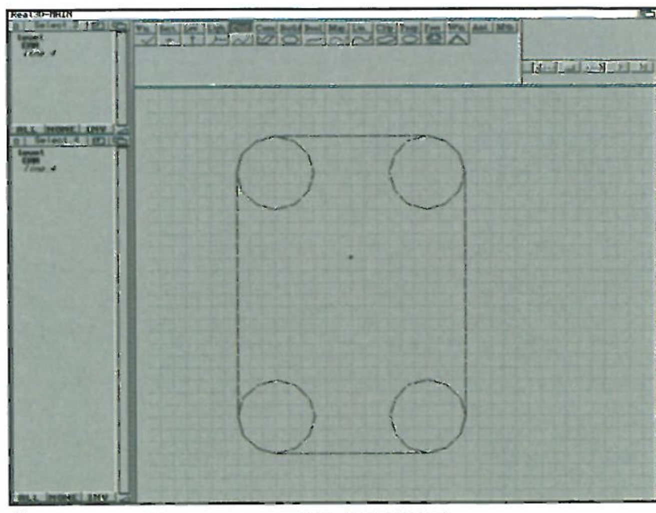
Pero imaginar ahora que el objeto requiere ser modelado mediante B-Spline. En tal caso, los bordes romos se consiguen mediante el acabado de las curvas de control en dicha zona con puntos de control dobles, o bien, con dos puntos de control cercanos. Mediante el uso de *MoveKnot Point* se procede a reposicionar los puntos de control que constituyen la zona roma hasta que la curva de la zona roma tenga las características del modelo original, y se consigue así de forma aceptable.

No obstante, este procedimiento es un método de modelar *a ojo*; pero se puede recurrir a otras diversas metodologías para acceder exactamente al tipo de modelado que se requiere en el momento. Si, por ejemplo, se observa que la zona roma es de características circulares, el método de crear la zona roma con un doble punto de control no sería ya exacto, dado que con esto se ofrece una curva B-Spline cuya forma es más parecida a la función $Y=1/X$ que a una curva circular. Además, el punto de control doble ofrece un borde romo de pequeñas dimensiones en com-





Creando el primer elemento de apoyo al modelado.



Curva creada sobre los elementos adicionales.

paración con la usual cercanía de los puntos de control en objetos bien definidos.

REALIZANDO BORDES ROMOS GRANDES

Si se dispone durante el modelado de un *Grid*, el usuario se asegura tanto de que las zonas de igual número de baldosas sean de igual longitud, como de conseguir ángulos básicos como el de 45 grados perfectos. A su vez, el hacer coincidir piezas en los ejes *X, Y* y *Z* se realiza de forma cómoda y eficaz.

La cuantización del espacio mediante un *Grid* tiene la limitación de que sólo se lleva a cabo mediante una malla rectangular. Se define una longitud para la longitud de la baldosa en horizontal y otra para la vertical, y es, entonces, cuando confiere cierta dificultad el realizar, por ejemplo, un borde romo de características circulares. Para llevar esto a cabo se realizará un uso combinado de *Drag* y de inserción de objetos de apoyo al modelado.

En definitiva, lo que hace falta es crear un círculo de la dimensión que se considere oportuna y acorde con el borde romo a modelar. Antes, habrá que decidir cuántos tramos se desea modelar en el borde romo. Por lo general, para una superficie B-Spline con 3 secciones de curva es suficiente aunque, a veces, se desea un modelo poligonal en el que se noten de por sí dichas secciones y se decida modelar 2, 3, 4 o más secciones.

Para poder realizar un borde romo de 3 secciones es necesario generar una circunferencia de apoyo de doce segmentos mediante *Create/Controls/CircularLine*. Se deberá realizar esta figura auxiliar con un *Grid* y teniendo cuidado de no rotar la figura en la creación (es decir, que para definir el tamaño se deberá desplazar el puntero sólo en *X* o sólo en *Y*).

Una vez creado el círculo, sus vértices nos servirán de apoyo al modelado de la caja, el cual será abordado mediante un *Build* de curvas de control. Ahora, se crea un rectángulo mediante *Create/Visibles/ Rectangle*, y se posiciona el círculo de apoyo en el interior del rectángulo de forma que los vértices superior e izquierdo del círculo toquen las líneas superior e izquierda del rectángulo.

Supuesto el caso de que las cuatro esquinas llevarán un borde romo de igual característica, se puede proceder a copiar el círculo creado y recolocar cada una de las copias a una esquina interna del rectángulo de igual forma que se procedió a hacer con el primero.

CREANDO LAS CURVAS DE CONTROL DEFINITIVAS

Hasta el momento, todo lo creado es meramente auxiliar y, por tanto, una vez creado el objeto deseado puede ser eliminado de la escena. Ahora es fácil desarrollar una curva en la que las zonas romas tengan exactamente el tamaño que se desea que tengan; por ello, hay que elegir *Create/Controls/B- SplineClosed* y proceder a introducir los puntos de control mediante *Drag* y aprovechar los puntos de control de las curvas auxiliares creadas.

Al emplear un *Build* en curvas de control, se podía repetir la última curva más veces

No sólo con este método se ha podido realizar una curva cuyo borde romo tenía el radio de curvatura que se deseaba sino que, como se habrá podido observar, es de fácil manufacturación una curva de control en la que sus zonas curvas tengan diferentes radios de curvatura elegibles por el usuario con anterioridad. Bastará con crear previamente distintos círculos, con diferentes tamaños, y recolocarlos convenientemente. Como se ve, hasta el número de secciones de la curva es independientemente elegible en cada zona curva con sólo determinar un número diferente de secciones a la creación de los objetos circulares de apoyo.

Es el momento de proceder a la creación del objeto 3D que en un principio era propuesto. Para ello, se ejecuta un *Build* de las curvas que se creen posteriormente. No obstante, se proponía una dificultad añadida en el caso, y es que debía tener en su zona superior dos huecos correspondientes a la supuesta zona del auricular y del micrófono. Para llevar esto a cabo, es bien conocido que no son recomendables los usos de las operaciones booleanas para este fin, debido a que cuando éstas son aplicadas a objetos 3D B-Spline, queda revelada la condición inevitable de objeto hueco debido a que los objetos B-Spline sólo son superficies infinitamente finas y envueltas en 3D.

Entonces pues, ¿qué hacer? El siguiente método de modelado es una fórmula genérica a tener a disposición en cualquier modelado. El concepto parte del hecho de que si existen dos objetos de igual forma y tamaño, colocados uno justo en la misma posición que el otro, a la hora del render ambos aparecerán como un sólo objeto. El "truco" es que, a partir de ahí, cada uno de estos es modelable por independiente.

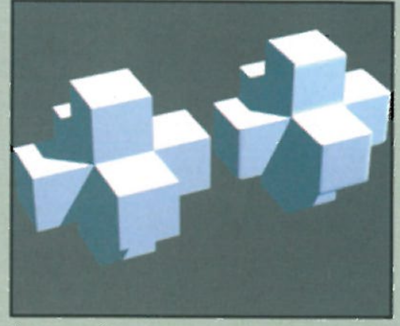
CREANDO EN 3D

En un principio, si sólo existiese un hueco, el caso sería más sencillo, debido a que al emplear un *Build* de las curvas de control, se podía repetir la última curva de control más veces achicando y reubicando éstas al tamaño y posición del hueco definido. Con lo cual, el hueco único es una especie de continuación de la propia creación del *Build*.

Pero esto no sirve para el caso de dos o más huecos, porque al realizar la primera, la superficie que parte desde la última curva de control que es extendida hacia el inicio de la creación del hueco taparía cualquier otra que hubiese.

Por lo tanto, para llevar esto a cabo, hay que tener dos objetos independientes en los que en cada uno se pueda definir el hueco propuesto, de forma que uno de los objetos no llegue a tapar el hueco del otro. Lo que se debe hacer en primera ins-

SOLUCIÓN AL EJERCICIO ANTERIOR


ILUSTRACIÓN DEL EJERCICIO ANTERIOR.

En el anterior número, se proponía un ejercicio de compleja y profunda comprensión con el fin práctico de que se ejercitase el ingenio y la inventiva del usuario. El ejercicio tenía la limitación de que debía ser realizado por dos objetos solamente y mediante la herramienta *Build*.

Para realizar la cruceta tridimensional de la izquierda era necesario desarrollar una serie de curvas de control, en las que cada vértice disponía de un punto de control triple. La sección del centro del objeto (visto de arriba hacia abajo) consta de doce vértices y cada uno con un punto de control triple. A su vez, cada sección debe ser duplicada tres veces también a fin de definir los cortes tajantes de las aristas. La sección pues realizada para la posterior construcción del *Build* debe ser desplazada marcando cuatro niveles de altura.

En el primer nivel las curvas de control extremas deben terminar por tener todos sus puntos de control confluidos en un solo punto para determinar el plano superior o tapa superior. Estas curvas de control contienen todos los puntos correspondientes a las secciones posteriores, emplazados en los cuatro grupos de puntos que constituyen las propias cuatro esquinas. Cada esquina pues contendrá nueve puntos de control: los tres propios de toda esquina, más dos grupos de tres correspondientes a la definición de los cubos salientes laterales.

En el segundo nivel, existen seis curvas de control: tres repetidas iguales a las tres anteriores, y otras tres en las que se despliegan todos los puntos de control concentrados en los cuatro grupos anteriores para conformar la forma de la sección central. La forma de finalizar la serie de curvas de control es sencilla. Basta repetir el proceso de forma inversa y continuando las secciones hacia abajo.

Para realizar la cruceta derecha con un solo *mesh*, se deberá realizar el mismo proceso anterior con la diferencia de que cada esquina tendrá sólo dos puntos de control, y las curvas de control en cada nivel se repiten dos veces en vez de tres.

tancia es dar volumen a la primera curva de control que fue definida mediante curvas auxiliares.

Ahora se deberá elegir una vista lateral o frontal, duplicar la curva en cuestión y desplazar las copias convenientemente para crear el *Build*. El objeto deberá ser cerrado con nuevas curvas de control que, estando a la misma altura que la última superficie, deberán ser cada vez más pequeñas, con objeto de crear una superficie plana en las zonas superior e inferior.

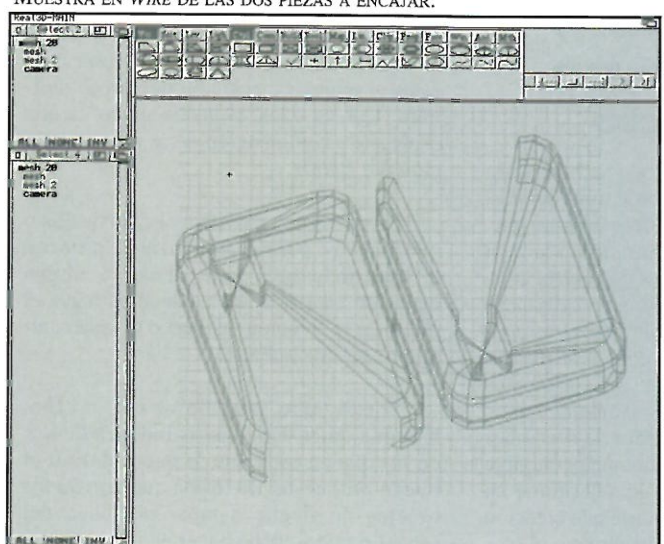
Teniendo un número de curvas igual o superior a cuatro, se eligen todas por orden en la ventana de selección y se ejecuta *Create/Freeform/BuildFromCurves*. Con ello se construye el lateral del objeto, con sus "tapas" o planos superior e inferior. En la "tapa" superior estará presente el primero de los huecos. Inmediatamente después se readaptan las curvas de control que acaban de ser usadas, de forma que las últimas curvas definan la forma de la oquedad restante (la inferior). Tras esto se volverá a ejecutar *Create/Freeform/BuildFrom Curves.*

Con ello se obtendrá un segundo objeto pero con la salvedad de que en éste el hueco se halla presente en su parte inferior. Ahora, pues, se divide el objeto en dos partes: una para el hueco superior, y otra para el inferior, y se recortan ambos de forma que el uno no tape la oquedad al otro. Si se procediese a realizar un render de lo que hay actualmente en la escena, éste revelaría un objeto base sin ningún tipo de hueco, ya que la superficie de uno taparía la del segundo, y viceversa.

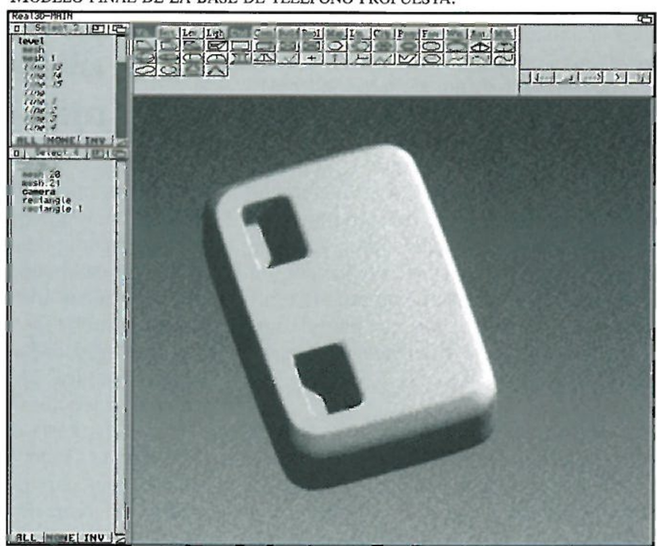
Esto se puede hacer de la siguiente forma: se elige el *mesh* creado el la ventana de selección y se ejecuta *Modify/FreeForm/ExchangeU&V* a cada uno de los dos *meshes* creados. Con ello se cambia la ordenación de la malla cuadrática que constituye cualquier *mesh,* de forma que los puntos en *X* pasan a ser puntos *Y* y viceversa. Todo ello ocurre sin que nada físicamente sea modificado; es meramente un cambio interno.

Ahora se ejecuta *Modify/Freeform/Surf.ToCurves* y se pasa de nuevo la superficie a curvas, pero con la salvedad de que, al haber sido cambiada la orientación U y V, las curvas de control resultantes de ese mismo objeto son las correspondiente a las intersecciones de las creadas originalmente, con lo cual, ahora se dispone de una serie de curvas de control longitudinal. Esto es lo que se perseguía en primera instancia, ya que para obtener el resultado deseado desde un principio, lo que se deberá hacer es construir la zona superior del objeto mediante un *Build* sobre las curvas de control que contienen el hueco superior y sin llegar a escoger todas las curvas, ya que habrá que desechar aquellas que pudieran tapar la oquedad contraria. Para construir la zona inferior del objeto se procede de forma similar: se seleccionan las curvas de control que contienen el hueco inferior y se desechan las últimas curvas (3 o 4) a fin de no tapar el hueco del objeto contrario.

MUESTRA EN *WIRE* DE LAS DOS PIEZAS A ENCAJAR.



MODELO FINAL DE LA BASE DE TELÉFONO PROPUESTA.



Si ahora se procede a realizar un render, ambos *meshes* coincidirán y encajarán plenamente tal y como si hubiese modelado solamente un objeto y, a su vez, presentará dos oquedades: una en su parte superior y otra en la inferior. Es posible que quede algún hueco no definido por ninguno de los dos objetos, lo cual deberá ser rellenado con un simple rectángulo.

OPERANDO CON VECTORES

Volviendo un poco atrás, en el punto en el que se definía la planta del objeto base introduciendo bordes curvos o romos mediante la ayuda auxiliar de círculos temporales, existe el caso en el que los círculos auxiliares se vean en la ventana de edición pequeños debido a que se tienen que visualizar todos los elementos de apoyo a la vez, en la misma ventana para crear, posteriormente, la curva deseada.

Tanto es así que incluso puede llegar a ser muy difícil la selección de los puntos mediante *Drag* debido a que con un *Drag* pequeño ya se pueden incluir dos puntos de control y tener un resultado anómalo. Para evitar posibles errores, en este caso se pueden hacer dos cosas. La primera, y más sencilla es, durante la creación de la curva de control, hacer click cerca de las zonas curvas tantas veces como puntos se deseen introducir, tanto si se hace el *Drag* como si no. A posteriori, se iría haciendo zoom a cada parte y reubicando con precisión cada punto fallido.

La segunda, más compleja pero más eficiente y elegante, es hacerlo bien desde un principio. Real3D dispone de lo que se denomina una "pila" de datos en la cual el usuario puede introducir coordenadas para ser usadas posteriormente.

Una pila básicamente es una lista de datos, de forma lineal y de un solo sentido. Es como una especie de saco, en el que se pueden ir metiendo cuantos datos se quiera. La regla es que el orden de salida de los datos es siempre inverso al orden de entrada (el primer dato en salir es el último que entró). Funciona como una pila de programación de ordenadores.

Con un *Drag* pequeño se pueden incluir dos puntos de control y tener un resultado anómalo

Existen diversas formas de introducir datos (los datos que se introducen son sólo y exclusivamente de carácter coordenada espacial) en la pila. Una de ellas es de forma directa, mediante la opción *Extras/Vectors/Enter*. Tras esto, aparecerá una ventana de requerimiento en la que se solicita la introducción de las coordenadas *X, Y, Z* del dato a introducir.

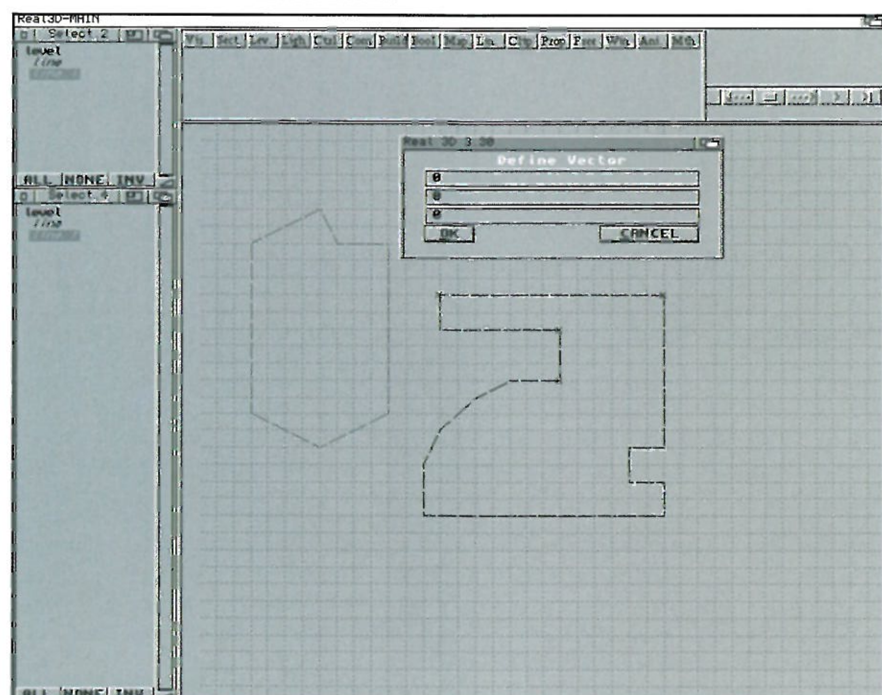
Otra forma más gráfica y funcional es la de realizar un *Drag* alrededor de un vértice existente, manteniendo mientras pulsada la tecla Shift, y se indica vértice porque pueden ser usados tanto los puntos de control habituales, como cualquier otro vértice que conforme el alambre de una primitiva por ejemplo.

De este modo, se puede, desde un principio, hacer zoom a cada esquina a modelar, e introducir por orden (como si se estuviese modelando ya la curva de control) en la pila de datos uno a uno los vértices a usar manteniendo siempre la tecla Shift pulsada. Una vez que han sido introducidos todos los datos, bastará sencillamente con proceder al modelado de la propia curva.

Ejecutar *Create/Controls/B-Spline Closed*, hacer click en la ventana de edición para activarla, aun poniendo un punto de control que no vale para nada, y ejecutar *Extras/Vectors/PullAll*. Tras esto, hacer click con el botón derecho del ratón para finalizar. Ya tendremos la curva deseada con un punto de más (el inicial para activar la ventana). Se elige igualmente con *Drag* y Shift, y se ejecuta *Modify/Freeform/ Delete* para eliminar el punto de control seleccionado.

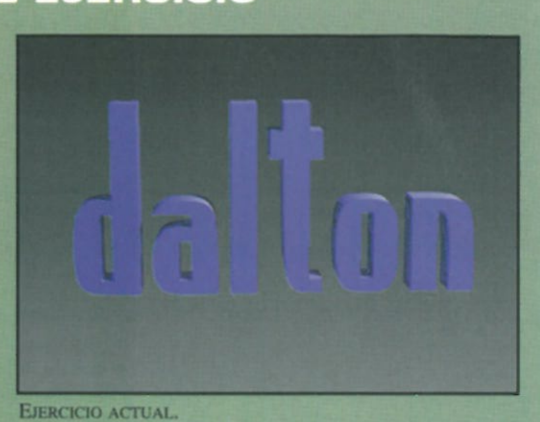
Un detalle a tener en cuenta es que, si tras introducir varios datos en la pila, se realiza un *Drag* sin tener la tecla Shift pulsada, Real3D usará todos los datos introducidos de forma automática y devolverá un único dato que corresponde a la media de todos los que hubiesen sido introducidos. Los datos usados son sacados para la operación y consiguientemente, perdidos.

INTRODUCCIÓN DIRECTA DE DATOS EN LA PILA.



EL EJERCICIO

En esta presente entrega se le propone al lector que realice el logotipo que se ve en la ilustración mediante una sola operación de la herramienta *Extrude* sobre sólo seis curvas de control. De esta forma, se recordará el modelado mediante *Extrude* y se activarán conocimientos pasados. Con ello, se rematará toda una jornada intensiva de modelado B-Spline.



EJERCICIO ACTUAL.

EN EL PRÓXIMO NÚMERO

En el próximo capítulo se retomará de nuevo la creación de materiales en Real3D entrando en nuevos parámetros definibles por el usuario, y determinando nuevas formas mapping que permiten un texturado más correcto o coherente con lo que se modela en determinadas ocasiones. Todo encaminado hacia una actitud profesional dentro del mundo de las 3D.



IMAGINE



Primeros pasos en el *Action Editor*

Autor: Miguel Angel Díaz

Nivel: Avanzado

En el *Stage Editor* se pueden cargar y colocar adecuadamente los objetos; los situamos ante la cámara, los iluminamos y preparamos la escena para la animación. Las animaciones simples se pueden crear con el *Stage Editor,* pero para animaciones más complejas el *Action Editor* es la pareja imprescindible del primero.

El *Action Editor* tiene las herramientas y el interfaz adecuado para controlar acciones complejas de los objetos a lo largo del tiempo por el que transcurre nuestra escena. En vez de ver los objetos tal y como son, en el *Action Editor* tenemos un historial del comportamiento de dichos objetos a lo largo de toda la escena; esto queda representado con diferentes barras de colores. En vez de seleccionar el objeto haciendo click sobre él, haremos click sobre las líneas de tiempo para que nos aparezcan nuevas ventanas en las que podremos introducir parámetros que influirán en la forma y el comportamiento del objeto que estemos modificando. Este método de proceder nos proveerá de un control más preciso sobre los objetos y sus movimientos.

Muchos de los comandos del *Stage Editor* también aparecen en el *Action Editor*, aunque este último tiene comandos únicos y efectos especiales que no están en el primero (también hay muchos elementos escénicos que se añaden desde el *Action Editor* y más tarde se manipulan desde el *Stage Editor*).

Aquí veremos cómo se trabaja con el *Action Editor*, cómo es su interfaz y cuáles son sus comandos para animar. Al igual que pasó cuando estuvimos "jugando" con el *Stage Editor*, las referencias cruzadas entre un editor y otro se hacen imprescindibles, por esto, hablaremos en más de una ocasión del *Stage*.

UN VISTAZO AL *ACTION*

Vamos a ver por encima las opciones que nos encontraremos en este editor en cuanto entremos en él. Vamos a acondicionarnos a su forma de trabajar y, más delante, veremos en profundidad las diferentes opciones.

En la figura 1 se aprecia la ventana principal sobre la que trabajamos. El *Action Editor* utiliza una especie de matriz donde se ponen los diferentes objetos de nuestra escena en horizontal, así como las acciones que éstos van a realizar a través de los diferentes frames en vertical. A su vez, la línea de la matriz que contiene el objeto está subdividida en otra serie de líneas horizontales en las que introduciremos diversos parámetros.

Observando la figura 1 se ve que las columnas que van de izquierda a derecha están numeradas del uno en adelante, a lo largo de la parte superior de la pantalla. Los objetos de la escena aparecen en el lado izquierdo de la ventana; primero la cámara, después los *Globals* y tras éste el resto de objetos en el orden que han sido cargados. Ya hemos comentado que las líneas en las que están encasillados los objetos están, a su vez, subdivididas en otras líneas. Estas sub-líneas son líneas de tiempo para parámetros como la posición, el tamaño, el alineamiento o los efectos especiales; se puede ver el título de cada línea a la izquierda de ésta.

Las líneas de colores que aparecen en mitad de la figura 1 son las líneas de tiempo. Por ejemplo, la primera línea que se ve empezando por arriba, la de color amarillo, es la que nos indica la posición de la cámara a lo largo de toda la escena. Hay líneas que son continuas, otras están divididas en tramos y otras sólo abarcan un cierto número de frames. Esto se debe a que hay parámetros que sólo afectan a una serie de frames, mientras que otros afectan a toda la animación.

Cómo el resto de editores de Imagine, el *Action Editor* también posee los típicos

comandos de edición que ya hemos visto en varias ocasiones: borrar, copiar, cortar, pegar, etc. En el cuadro de iconos que aparece en este artículo se pueden ver los iconos que los representan y qué hacen cada uno de ellos.

Cuando se hace doble click sobre una de las líneas de tiempo, aparece una ventana. Por ejemplo, en la figura 2 se aprecia la ventana que resulta tras hacer doble click sobre la barra *Align* de la cámara y se puede cambiar su alineamiento con sólo entrar nuevos valores en los casilleros de los ejes.

En este apartado hemos visto por encima cómo es el *Action Editor*; en los siguientes vamos a entrar en profundidad en todas sus opciones.

LA VENTANA DEL *ACTION EDITOR*

Antes de poder entrar en el *Action Editor* hay que tener abierto un proyecto. Ya vimos hace unos números cómo se iniciaba un proyecto desde la versión PC y cómo desde la versión Amiga (en este punto es donde más diferencias hay entre una versión y otra). Un proyecto no es más que un grupo de objetos que hemos cargado en el *Stage Editor*, los hemos colocado adecuadamente y los hemos animado.

Cuando se entra por primera vez en el *Action Editor*, tendremos el número de frames que hayamos asignado cuando dimos de alta el proyecto. Si este número no era el correcto, ahora es el momento de rectificar. Definir el tamaño de la animación antes de cargar objetos en el *Stage Editor* es algo bastante importante, ya que si lo hacemos a posteriori, desde el *Action Editor*, podemos encontrarnos con problemas: cuando cargamos un objeto en el *Stage Editor* ya sabemos que éste aparecerá en la escena desde el frame que fue cargado hasta el final de la animación, pero si ésta es ampliada desde el *Action Editor*, el objeto no estará presente en los nuevos frames que han sido añadidos. Aunque se puede hacer que los objetos aparezcan en los nuevos frames, se trata de una labor bastante desagradable ya que hay que realizarlo objeto por objeto.

Para decidir cuántos frames debe tener nuestra animación, primero debemos saber para qué elemento de salida está destinada la animación. Si, por ejemplo, estamos haciendo una animación que va a ser reproducida en un vídeo PAL, ésta tendrá que estar compuesta por 30 frames por segundo; sin embargo, hay programas multimedia que reproducen las animaciones a 15, 20 o 24 frames por segundo. Con un simple cálculo matemático sabremos el tamaño de la animación: multiplica los frames que ésta debe tener en cada segundo por el número de segundos que va a durar la animación.

• *Scroll* de la ventana del *Action Editor*
Cuando el número de frames es demasiado alto para que quepa en la ventana, éstos se pierden por la parte derecha.

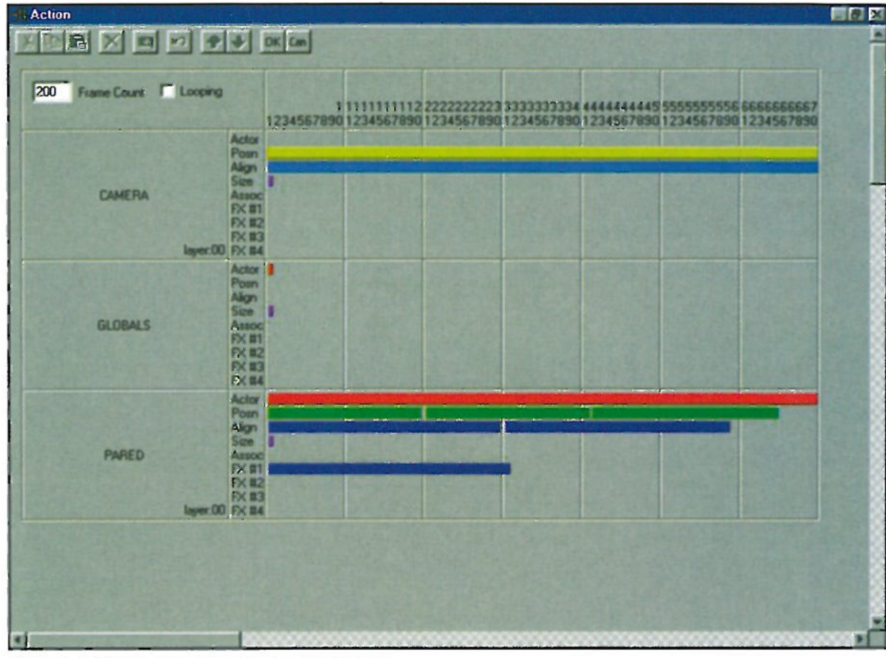


FIGURA 1. VISTA GENERAL DEL *ACTION EDITOR*.

Para tener acceso a los frames que están más allá de esa frontera, en la parte inferior hay un deslizador que permite llegar a la zona oculta. Igualmente pasa cuando la lista de objetos es más larga que lo que cabe en la ventana; aparece un deslizador en la parte derecha que nos permite subir y bajar por la lista de objetos.

•La lista de objetos y los parámetros
A lo largo de la parte izquierda de la ventana podemos ver la lista de todos los objetos que componen nuestra escena (a los objetos también se les llama actores ya que los primeros actúan como éstos en una escena). Esta lista incluye objetos, cámara y luces, además de una entidad especial llamada *Globals* que controla todos los parámetros globales de la escena. También sabemos que los objetos se cargan en capas (*layers*) desde el *Stage Editor*, así que el número de la capa donde está incluido el objeto aparece bajo el nombre de éste (por ejemplo: *layer:00*).

La lista de parámetros que aparecen a la derecha del nombre del objeto definen su nombre y localización, su posición, orientación, tamaño, su relación con otros objetos y efectos especiales relacionados con él. Más adelante explicaremos estos parámetros en profundidad pero, de momento, vamos a dar una breve descripción de ellos:

• ***Actor***: Define el directorio donde se encuentra el objeto y el nombre del archivo, el primer y último frame, parámetros que afectan al *morphing* del objeto y a movimientos cíclicos, el State actual en el que se encuentra, atributos de propiedades lumínicas y los parámetros globales completos. Esta descripción no es válida para la cámara.

• ***Posn***: Define la posición del objeto en el mundo 3D a lo largo de toda la escena.

• ***Align***: Define la rotación del objeto a lo largo de la animación.

• ***Size***: Define el tamaño del objeto durante la animación.

• ***Assoc***: Causa que el objeto quede pegado o asociado a otros objetos y mantenga su distancia relativa, rotación, etc.

• ***F/X***: Efectos especiales. Puedes definir hasta cuatro efectos especiales para cada objeto.

•Las barras de tiempo
Cada actor tiene una serie de barras codificadas por colores, asociadas con sus parámetros *Actor*, *Posn*, *Align*, *Size* y *FX*. En la figura 1 se ve cómo algunas barras ocupan toda la animación y otras sólo algunos frames; esto es porque representan de una forma visual qué pasa con el objeto entre los *Key Frames*.

Los *Key Frames* están situados al principio y al final de cada uno de esos tramos de las barras de tiempo, e Imagine añade un pequeño espacio en blanco entre tramo y tramo de estas barras (fíjate en la barra *Posn* del objeto PARED). El primer frame siempre es un *Key Frame*, por eso siempre aparece una barra en el frame 1 seguida del resto del espacio en blanco.

Las barras de tiempo aparecen en cuanto el objeto se carga desde el *Stage Editor* y, como ya sabemos, los objetos pueden ser cargados desde cualquier frame. El frame donde el objeto es cargado, automáticamente, se establece como el primer *Key Frame*. Debido a que el último frame de la barra de tiempo es también un *Key Frame*, éste define el valor de los parámetros para los siguientes frames hasta el final de la animación o hasta que otro *Key Frame* aparezca.

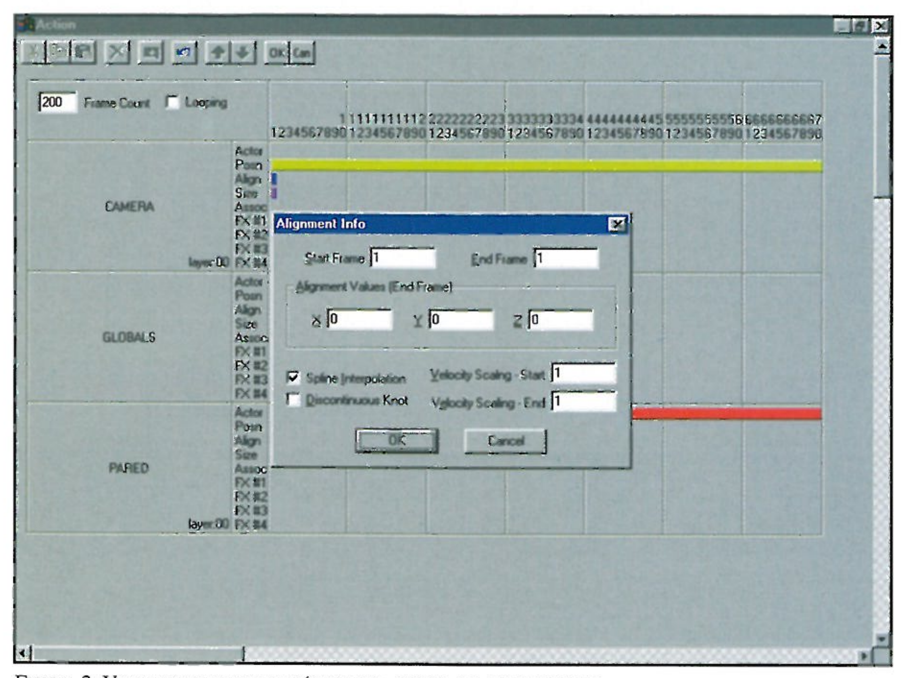


FIGURA 2. VENTANA DE INFORMACIÓN DE UNA BARRA DE ALINEAMIENTO.

Cada barra de tiempo tiene un color diferente para ayudar a distinguirla fácil y rápidamente. El color de la barra también indica un cambio en la acción del objeto durante el transcurso de un número de frames especificado. Por ejemplo, un barra amarilla indica que el objeto está siguiendo un *path*. Aquí tienes el color asociado a cada parámetro y su significado:

• BARRA *ACTOR*
Puede ser de dos colores diferentes: **Roja**, que define los parámetros específicos del tipo de objeto del que se trate. La barra se extiende desde el primero hasta el último frame en los que el objeto tiene presencia, por lo tanto, en los frames en los que esta barra no está presente significa que el objeto no aparece en la escena. La cámara no tiene barra *Actor* ya que no puede ser borrada de la animación. O **Naranja**, que indica que el objeto sufre una transformación en su geometría durante los frames que se especifican (un *morphing*).

La ventana del *Action Editor* está pensada para ver la escena en organigrama

• BARRA *POSN*
Consta, también, de dos colores: **Verde,** el objeto está cambiando de posición gradualmente de un *Key Frame* a otro (del primer frame de la barra hasta el último frame de esta) y **Amarilla,** en donde el objeto está siguiendo un *path* durante la duración de la barra.

• *ALIGN*
Los diferentes colores de este tipo de barras son: **Azul,** el objeto está rotando entre dos key frames; **Celeste,** el objeto está alineado a un *path*; y, finalmente, **Azul claro,** el objeto está alineado respecto a otro objeto.

• *SIZE*
Esta barra sólo consta de un color, el Morado, en donde el objeto está cambiando de tamaño.

• BARRA *ASSOC*
Igualmente, este tipo de barra consta de un solo color, **Rosa**, el objeto está pivotando en un arco alrededor de otro objeto.

• *F/X*
Para finalizar, la Barra F/X consta del color **Azul**, que significa que se está aplicando un efecto especial a lo largo de los frames que se especifican. Estos efectos especiales los veremos más adelante.

PRIMER EJERCICIO EN EL EDITOR DE ACCIONES

Los objetos se cargan normalmente desde el *Stage Editor*, pero si lo añadimos a la escena desde el *Action Editor* podremos emplazarlo y manipularlo con mayor precisión. Si miras en la columna de actores te darás cuenta que la última línea se llama *New* y se usa para añadir nuevos objetos a la escena. Cualquier tipo de objeto se puede añadir desde la línea *New*, incluido un foco de luz:

• Hacemos doble click sobre la barra *Actor* de *New*.

Desde el *Action Editor* podemos variar el nombre que internamente tiene cada objeto

• Aparecerá una ventana como la de la figura 3 donde se pregunta si el objeto es uno normal, un foco de luz o un eje.

CUADRO 1. ICONOS QUE APARECEN EN EL *ACTION EDITOR*

1. **Cortar**: Realiza una copia del elemento seleccionado en la memoria del ordenador y además borra la copia actual. Esto se hace para, más tarde, realizar una operación de pegado en otro sitio.
2. **Copiar**: Desarrolla una copia del elemento seleccionado en la memoria del ordenador, pero sin borrar el original.
3. **Pegar**: Pega en el lugar seleccionado el elemento que antes ha sido cortado o copiado a la memoria.
4. **Borrar**: Borra el elemento que tengamos seleccionado.
5. **Propiedades**: Nos facilita información sobre el elemento que tenemos seleccionado.
6. **Eliminar última acción (Undo)**: Pulsando sobre este icono anularemos la última acción que hayamos realizado.
7. **Subir/Bajar**: Estas flechas hacen que la lista de actores suba o baje.
8. **Ok**: Sale del *Action Editor* dando por buenas las acciones que hayamos llevado a cabo y salvando estos cambios.
9. **Cancel**: Sale del *Action Editor* anulando todos los cambios que hayamos realizado.

- Si aceptamos la opción *Normal Object* se nos invitará a que carguemos el objeto desde el disco duro.

- Tras cargar el objeto, y tras introducir en qué *layer* queremos colocarlo, ya nos aparece la ventana estándar de información del objeto donde podremos especificar en qué frames queremos que aparezca, en qué posición, etc.

- La barra de color de *Actor* debe aparecer en cada frame en el que el objeto deba estar presente. Las barras *Posn*, *Align* y *Size* sólo necesitan estar presentes en el primer frame; sus parámetros estarán intactos hasta que se cree un nuevo *Key Frame*.

- El objeto ya está en nuestra escena.

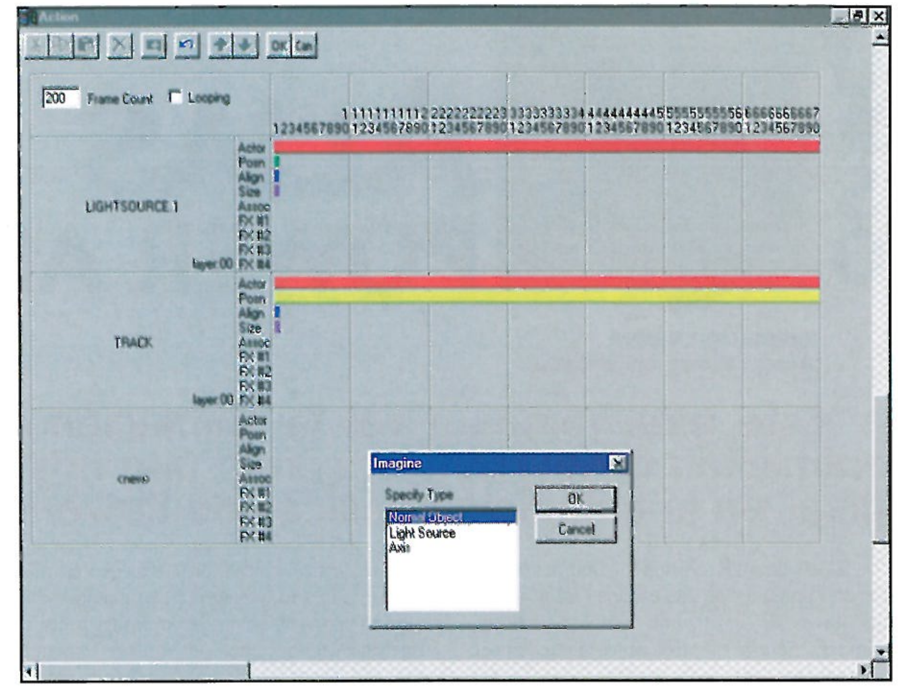


FIGURA 3. VENTANA QUE APARECE CUANDO SE AÑADE UN NUEVO ACTOR.

AÑADIENDO UN *KEY FRAME* EN EL *ACTION*

Ya vimos cómo podíamos añadir un *Key Frame* desde el *Stage Editor* y ahora vamos a ver cómo hacerlo desde el *Action Editor*:

- Elegimos el objeto en el que quieres crear el *Key Frame* entre los de la columna de actores del *Action Editor*.

- Pulsa con el botón izquierdo del ratón sobre la barra en la que quieras crear el *Key Frame* y, sin soltar dicho botón, arrastra la línea hasta el frame que quieras que dure dicha acción. Esta acción también puedes realizarla haciendo doble click sobre la barra seleccionada y luego ajustar los frames en los parámetros de la ventana que aparece posteriormente.

- Tras esta última operación aparecerá una nueva ventana que variará dependiendo del tipo de barra de tiempo en donde estemos creando el *Key Frame* (posición, alineamiento, tamaño, etc.) y del tipo de objeto del que se trate (normal, foco de luz o eje).

Una rejilla ordena las filas para los objetos y las columnas para los *frames*

- Después de realizar estas operaciones y de aceptar la ventana con los parámetros, podemos observar cómo ha aparecido un nuevo fragmento de barra en los frames que hemos seleccionado.

VENTANA DE INFORMACIÓN DE ACTORES

Cuando hacemos doble click sobre el nombre del actor en cuestión aparece una pequeña ventana que nos permite cambiar el nombre del objeto y el número de la capa (*layer*) en la que estará.

La habilidad de poder cambiar el nombre del objeto es algo magnífico. Cuando se cargan los objetos en Imagine, éste utiliza sus nombres internos para colocarlos en la columna izquierda. Si dos objetos tuvieran el mismo nombre, cómo, por ejemplo, un foco de luz, un eje o un *path*, Imagine añadirá un índice numérico al nombre del objeto. Por ejemplo, los ejes se llaman por defecto AXIS, pues Imagine los llamaría AXIS, AXIS 1, AXIS 2, etc. Esta denominación puede ser bastante confusa y, por eso, poder cambiar estos nombres es algo simple en apariencia pero bastante valioso: en vez de tener tres focos de luz que se llamen LIGHTSOURCE, LIGHTSOURCE 1 y LIGHTSOURCE 2, podemos cambiar estos nombres por FOCO PRINCIPAL, FOCO TRASERO, FOCO ELEVADO.

El otro elemento importante que podemos variar desde esta ventana es la capa (*layer*) en la que está el objeto. Se puede modificar la capa en la que encontraremos este objeto introduciendo, simplemente, otro número en el casillero. El uso de las capas nos permitirá organizar mejor los objetos de nuestra escena y, en su momento, ya comentamos el uso más adecuado que podíamos darle a esta herramienta.

EN EL PRÓXIMO NÚMERO

Hemos visto de una forma bastante genérica todas las posibilidades que tenemos desde el *Action Editor*. Nos hemos centrado en el manejo superficial de este editor, en los comandos que nos encontramos nada más entrar en él.

En el siguiente capítulo de este tutorial vamos a ver todas las ventanas y opciones que cuelgan de las barras de tiempo y aprenderemos a utilizar todos los parámetros que contienen. Incluso es posible que lleguemos a ver algunos de esos efectos especiales de los que hemos hablado en varias ocasiones.



CURSO

ALIAS POWER ANIMATOR

SGI

Animación básica
Autor: Bruno de la Calva

Nivel: Básico
Plataforma: SGI

Este mes vamos a ver la animación a través de claves de animación o KeyFrames, con lo que nos meteremos a profundizar en la animación básica con Power Animator.

El modo más común de animar todos los componentes de una escena es por medio de claves de animación. Luces, texturas, cámaras, *Shaders* y otros objetos pueden ser controlados con este método. La forma genérica de actuar es la siguiente: primero se verifica que el *Time Slider*, barra con la que se controlan los tiempos en la escena, aparece en pantalla. Si no es así, en el desplegable donde están las herramientas de animación, *Animation*, se encuentra *Toggle Time Slider*. Una vez solucionado esto, se ha de activar el objeto sobre el que se está trabajando. En *Animation* también está *Set Keyframe* de modo que, haciendo click en él, el objeto que esté activado tendrá asignada su primera referencia.

Con el ratón se colocará la barra de tiempos para que, a continuación, se le aplique al objeto la transformación que sea necesaria. Esta transformación puede ser una simple rotación, una variación en el parámetro que controla la intensidad de una luz, un cambio de color en un *Shader*, etc. Todas los parámetros susceptibles de ser animados están detallados en *Param Control*. Ahora, sólo quedaría determinar el *Keyframe* correspondiente acudiendo de nuevo a *Set Keyframe*.

En la parte superior de la pantalla se encuentran los controladores con los que se puede visualizar el resultado de la operación. También existe otra alternativa, que es más aproximada a lo que sería la animación si se hubiera lanzado el *render* definitivo. Es la función *Playback* que, lógicamente, está ubicada en *Animation* y que vendría a ser un *Render* del *Wire* (modelo en modo de alambre), sin luces y sin texturas o, en modo *Quick Render*. Al igual que *Param Control*, *Playback* y *Playback Options* tendrán su espacio.

Los objetos animados en cualquiera de sus parámetros son fácilmente diferenciables en la *SBD*. El nodo que corresponde a cada uno de ellos aparecerá inclinado hacia la derecha. En el caso en el que no estén animados, el nodo será rectángular.

En el caso de *Shaders*, texturas y luces, el rectángulo inclinado aparecerá tanto en el *Multi-lister*, al lado del nombre, como en el interior de su menú, apareciendo en el *display* que está animado. Algo muy funcional es la posibilidad de poder visualizar la animación de cualquiera de estos componentes desde el *Multi-lister*. Basta solamente con hacer *click* en el rectángulo inclinado para completar dicha visualización. Por otro lado, en la ventana de perspectiva, las cosas funcionan del mismo modo.

Como ya se ha dicho, el objeto que se quiera animar ha de estar activado con anterioridad a fijar el *Keyframe*. Sin embargo, en *Shaders*, texturas y luces, hay que seleccionar qué parámetros van a ser animados y cuáles no lo serán en el *Param Control*. Más tarde, se irán variando los valores de cada *display* en función de la animación desde dentro del menú correspondiente.

Hay que puntualizar que las luces también pueden ser transformadas como si fueran geometría. Por tanto, la rotación, la translación y el escalado de una luz es animada análogamente en conjunto, actuando sobre el nodo superior, o por componentes, sobre cada uno de ellos.

Los esqueletos también se pueden animar con *Set Keyframe* haciendo uso de la cinemática inversa. Lo primero es activar el controlador final del esqueleto. En el menú se ha de seleccionar la opción correspondiente a esqueletos, *Sets*, para que, una vez colocados unos cuantos *Keyframes*, se pueda usar *AutoKeyframe*, con la que sólo se darán *Keyframes* a aquellos parámetros que hayan alterado sus valores.

En el artículo del mes pasado se habló de los *Constraints* y de su función. Para aquel que no haya leído dicho artículo, recordar que los *Constraints* son objetos que se asignan a estructuras más complejas con el fin de simplificar la animación de las mismas. Es decir, aunque la animación recaería sobre el total de esa estructura, por ejemplo, un esqueleto, tan solo se tendría que animar este *Constraint*. Pues bien, una vez seleccionado un *Constraint*, se ha de acudir a *Param Control* para comprobar si los parámetros que controlan el *Constraint* están activados. Pulsando *Set Keyframe* tendrá la primera clave.

En el módulo de animación básica todo lo referente a *Clusters* no aparece, ya que éstos pertenecen al módulo avanzado. En cualquier caso, es importante mencionar su existencia, dada su estrecha vinculación al apartado de esqueletos e *Ik Handles*.

Como ya se ha señalado, la ventana de perspectiva se puede animar. Desde esta ventana existe la posibilidad de usar varias cámaras como si de un control de realización se tratase. Para crear una nueva cámara

se ha de abrir el menú *Layouts*, en el cual aparecerá *New Camera*.

Para animar las distintas cámaras, la ventana de perspectiva tiene que estar activada previamente. Como siempre, es necesario ir a *Param Control* para comprobar que lo que se va a animar está "autorizado". Dentro de *Param Control* está el icono donde se podrá restringir o no la animación de las cámaras, y es muy importante que la tangente de salida de la curva de animación de la cámara se coloque en modo *Step*. Aunque esto se puede arreglar más tarde en *Action Window*, desde la función *Set Keyframe* es posible determinar con anterioridad el tipo de tangente, tanto de entrada como de salida.

El porqué de esta reseña es lo siguiente. En el momento en el que existan varias cámaras en la escena, el paso de una a otra se produce en el intervalo de un *frame*. Es por esto por lo que las tangentes nunca podrían ser ni *Smooth*, ni *Slow*, ni ninguna de las otras que existen como posibilidad.

Usando el *Time Slider* para establecer los tiempos y el pulsador con el que se selecciona la cámara que se quiera animar, se irán creando los distintos movimientos de cámara y también todos los cambios de cámaras que sean necesarios para obtener el resultado deseado.

SET KEYFRAME OPTIONS

Las distintas opciones que ofrece *Set KeyFrame* han ido apareciendo a lo largo de todo este tutorial. Estas opciones se reducen a cinco: *Objects*, *Parameters*, *Hierarchy*, *Frame* y *Tangent Type*.

Siguiendo el orden de la enumeración anterior, *Objects* va a ser la primera a analizar. Dentro de *Objects* están *Active* y *Sets*. Si se está en el modo *Active*, la función recae directamente sobre los objetos que estén activados. Un *Cv*, un polígono, un *Image Plane*, una luz, o la cámara pueden ser seleccionados con *Pick Objects* para darles un *Keyframe*. En el caso de los *Constraints*, éstos están activados si el esqueleto al que están referenciados lo está. De igual modo, el *Cluster* estará activado si el *CV* asociado lo está. Luces, *Shaders*, texturas o *Enviroments* son considerados activos cuando lo están en el *Multi-lister* y, aunque en el *Multi-lister* hay siempre un elemento activado, éste no será animado en ninguno de sus parámetros mientras no se le haya marcado en el *Param Control*. En cuanto a las ventanas, mientras se esté trabajando en una de ellas será ésta la que esté activada. *Sets* es la segunda alternativa de este apartado. Los *Keyframes* recaerán sobre todos aquellos objetos que estén en los *Sets* seleccionados en la ventana *Set Lister*.

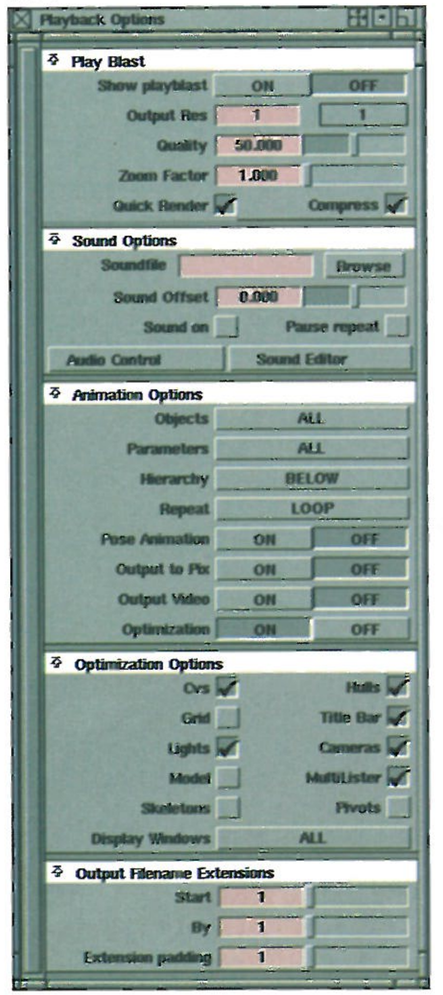
Los tipos de variación de parámetros a los que se puede optar son *ALL* (el *Keyframe* afectará a todos los parámetros animables del objeto) y *GLOBAL/LOCAL* (donde el *Keyframe* es aplicado sólo a aquellos parámetros que se han determinado previamente en el apartado que éstos tienen en *Param Control*).

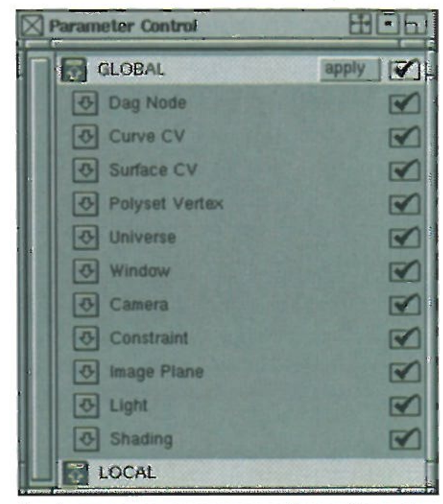
Las jerarquías posibles son: *NONE*, el *Keyframe* es asignado a los objetos que estén seleccionados; *ABOVE*, objetos seleccionados y todos aquellos que estén por encima de éstos; *BELOW*, objetos seleccionados y todos aquellos que estén por debajo de éstos y, por último, *BOTH*, que unifica *Above* y *Below*.

Si se usa la opción *Below*, los *Keyframes* se aplicarán al nodo seleccionado y las geometrías que correspondan a esta restricción. Sin embargo, los *CVs* que pertenezcan a las citadas geometrías no se verán afectados en ningún caso. Si se quisiera dotar de *Keyframes* a los *CVs*, habría que seleccionarlos desde la malla que los contiene y animarlos de forma independiente. Por otro lado, en el caso de objetos, *CVs*, polígonos, *Image Planes* y cámaras, la jerarquía se refiere a la jerarquía representada en la *SBD*. En *Shaders*, luces y texturas, la jerarquía está implícita en el *Multi-lister*.

Con *FRAME* se puede elegir la manera de colocar en el tiempo los *Keyframes*. *Current* coloca los *Keyframes* en el "tiempo actual". Este "tiempo actual" es el que viene indicado en la ventana donde se esté trabajando, en el *Time Slider* en *Action Window*. Con *Prompt*, el sistema propone introducir el tiempo en el que ubicar el *Keyframe* justo en el momento en el que se acude a *Set Keyframe*.

MENÚ DE *PLAYBACK*.





PARÁMETROS ANIMABLES DESDE *PARAM CONTROL*.

La última alternativa que oferta el menú de la función *Set Keyframe* es la referente a las tangentes. Para cambiar el tipo de tangente a aplicar, basta con hacer *click* en cualquiera de los dos *displays* que aparecen en el desplegable. Las tangentes de las curvas en cada punto describen cómo es interpretada la acción entre los *Keyframes* que la conforman. Cada *Keyframe* tiene dos tangentes, una de entrada y una de salida, y determinan el comportamiento de la acción desde la izquierda, con la tangente de entrada, y desde la derecha, con la tangente de salida (en el cuadro adjunto a este artículo podemos ver los tipos de tangente que existen. ✍

TIPOS DE TANGENTE

Smooth: Crea una transición suavizada. Las tangentes son co-lineales, es decir, ambas con el mismo ángulo, con lo que se consigue que no haya sacudidas en esa parte.

Linear: Crea una línea recta entre dos *Keyframes*. El segmento que preceda o que suceda a un *keyframe* con tangente linear será también una línea recta.

In/Out: Equilibra la salida de la curva respecto del *Keyframe* anterior o del posterior.

Flat: Hace que la caída tenga grado 0.

Fast: Si la tangente de entrada está en modo *Fast*, el movimiento se irá acelerando desde el primer *Keyframe* hasta el siguiente. Si la tangente es de salida, el movimiento será de deceleración.

Slow: En la tangente de entrada provocará un retardamiento del movimiento hasta el siguiente *Keyframe*. En la tangente de salida, en cambio, hace el movimiento más lento al principio del segmento de la curva.

Step: Hace que el paso de un *Keyframe* a otro sea horizontal. Esta opción es muy útil para animar *booleanas*, cambios de cámara, etc. Los valores por encima de 0.5 indican que la función está activada.



STRATA STUDIO PRO

MAC

Animación avanzada
Autor: Fernando Cazaña

Nivel: Básico

Este mes terminaremos de explicar las últimas funciones de Strata Studio Pro 1.75, y el mes que viene comenzaremos con la versión 2.0, por lo que finalizaremos con las funciones avanzadas de animación.

Tras explicar todas las funciones de Strata Studio Pro, en el próximo número comenzaremos a explicar la versión 2.0 de este programa, en la cual se han modificado todo el interfaz y se han añadido gran cantidad de nuevas funciones, como puede ser las partículas o las luces volumétricas.

Este mes veremos todas las funciones de la animación avanzada, que son las últimas que nos quedan por examinar, y nos permiten gran variedad de efectos para la animación.

Todos estos efectos se encuentran en el menú desplegable de animación, situado en la parte superior de la pantalla. Pinchando sobre él aparecen al final de este menú, y son los de *Morph, Path Filters, Physical Effects, Special Efects* y *Warp*.

MORPH

Este efecto es una forma de transformar un objeto en otro distinto de una forma suave y animada. En Strata hay dos formas de realizarlo:

La primera de ellas es una manera un poco diferente de realizar el *Morph*; el efecto lo realiza cogiendo el primero de los objetos y realizando una separación de todos los polígonos, los modifica para adaptarlos al segundo modelo y, al final, los vuelve a juntar en una malla completa. Esta forma de realizarla se llama *Particle Morph* y pulsando sobre ella nos aparece una ventana, la cual está dividida en dos partes. En la primera nos aparece un listado de todos los objetos que tenemos en la escena, en la cual seleccionaremos el primer modelo para realizar el *Morph*. En el apartado de la derecha también nos aparece un listado, igual que en el anterior, para seleccionar el segundo modelo, que será el objeto que permanezca desde ese *frame* y debajo de estas dos partes podemos indicar en cuánto tiempo queremos que se realice el *Morph*. La segunda forma de realizarlo es que, en todo momento, sea un objeto completo, no como en el caso anterior, que se convertía en polígonos separados.

PATH FILTERS

Este apartado tiene tres funciones distintas, pero todas son referidas a la función esencial, que es la de *Convert to Path*.

Lo primero que tenemos que saber es qué es un *Path*; como la palabra dice es un camino o recorrido que un objeto realiza en el tiempo. Cuando tu animas un objeto, podemos realizarlo de dos formas distintas: una de ellas es moviéndolo y dando *Keys*, y la otra creando una curva y asignándosela como camino a un objeto, pues *Convert to Path* realiza esto último.

Para utilizar esta función necesitamos primero dos cosas; dibujar una curva para utilizarla como camino, y convertir nuestro modelo a un *Shape*. Una vez realizado esto, pincharemos sobre la función de *Path Filters/Convert to Path* con la curva seleccionada; nos aparece una ventana en la que se pueden seleccionar un *Shape* de entre todos los de la escena, así como indicar cuándo empieza la animación y la duración de la misma en segundos.

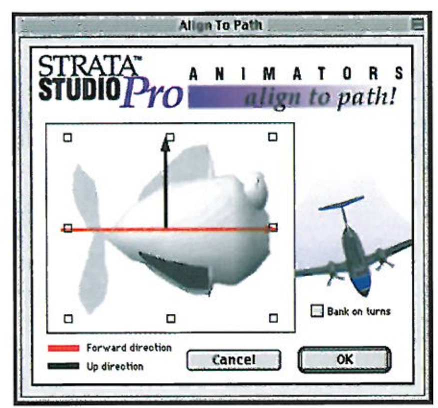
La siguiente función es la de *aling to path*; ésta nos permite definir la parte del objeto que estará delante y la parte superior del modelo para que se mueva correctamente. Por último podemos definir si en las curvas se balancea el modelo o no.

La última función es la de *Point at,* que nos permite que las cámaras de la escena estén constantemente mirando al objeto que hemos animado, por Path. La ventana que nos aparece esta dividida en dos partes, en la primera seleccionamos la cámara que queremos utilizar, y en la segunda el objeto al que queremos que mire, y debajo de éstas podemos decirle que la siga todo el tiempo sólo cuando se mueva o que empiece en un segundo determinado y termine en otro distinto.





VENTANA DE EDICIÓN DE *ATOMIZE*.



AQUI VEMOS CÓMO ALINEAR UN OBJETO A UN *PATH*.



EFECTO DE DESCOMPOSICIÓN DE UN MODELO.

PHYSICAL EFFECTS

Seguramente el lector se habrá preguntado alguna vez cómo realizar una pelota botando. Pues bien, este efecto hace exactamente eso. En la ventana que nos aparece-podemos ver tres *scrolling* en los que se podrá definir, primero, la altura desde la que cae un objeto, la gravedad que existe en el mundo virtual y la cantidad de rebotes también nos permite optimizar la animación de los rebotes al número de fotogramas por segundo que tenga nuestra animación, y la duración total de dichos rebotes.

SPECIAL EFECTS (EXPLODE)

Esta función que tiene Strata nos posibilita el poder realizar explosiones de modelos, pero lo que realiza esta función es dividir el modelo en polígonos y expandirlos en el espacio sin ningún efecto de fuego o similar.

En la ventana que aparece al activar esa función vemos tres manejadores que nos permiten cambiar los parámetros de la fuerza con que explotará el objeto, la gravedad existente y el tiempo de vida de la explosión. También nos posibilita introducir la duración total de la explosión.

SPECIAL EFECTS (ATOMIZE)

Esta otra función nos permite convertir un modelo en esferas como si estuviera compuesto de átomos. Al activarla, nos aparece una ventana parecida a la de los demás efectos en la cual, mediante unos manejadores, podemos definir la cantidad de detalles que tendrá nuestro modelo al convertirse en esferas, la estabilidad que tendrán, si estas esferas se moverán, la energía que tendrán las esferas y la vida, así como permitirnos especificar la duración de la transformación de objetos sólidos a atomizados.

SPECIAL EFECTS (SHATTER)

Shatter es un efecto que permite que cualquier modelo se deshaga en polígonos y éstos caigan por el efecto de la gravedad. En la ventana que nos aparece al seleccionar el efecto se puede especificar la turbulencia que tendrán los polígonos cuando estén cayendo, que girarán por el efecto del aire. También nos permite definir la gravedad que exista en nuestra animación y el tiempo de vida del modelo, antes de deshacerse, y la duración de toda la animación.

WARP

Este efecto posibilita el poder retorcer un modelo en los tres ejes, es decir, en el eje vertical, en el horizontal y sobre sí mismo. Esta herramienta es tanto de modelado como de animación, puesto que se puede especificar que el efecto lo realice en un tiempo determinado. En la ventana de edición de este efecto podemos definirle el número de grados de rotación, el ángulo de subida y el número de grados sobre el que se retorcerá el mismo. En el apartado *Expert* nos permite especificar la base de la circunferencia final, mientras que en la parte inferior de la ventana podemos activar que se anime y la duración de dicha animación.

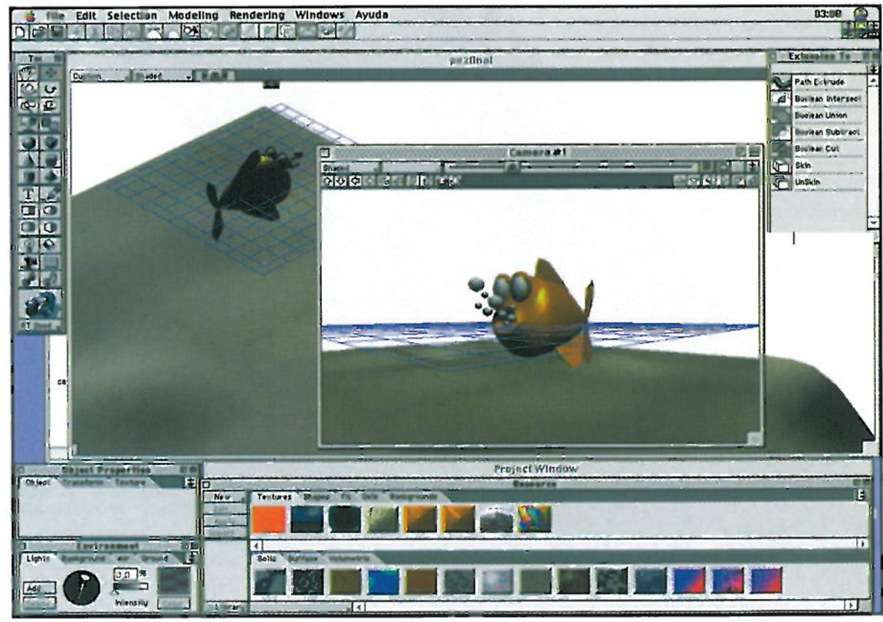
A PARTIR DE AHORA...

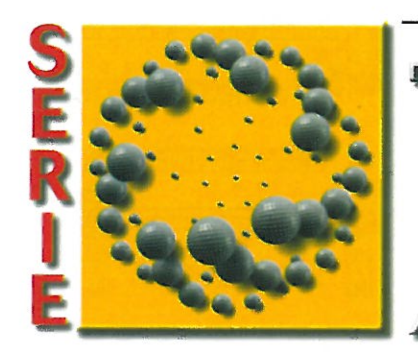
Con estos efectos terminamos de ver todos los parámetros de Strata Studio Pro 1.75, por lo que el mes que viene comenzaremos con la siguiente versión de Strata, que está totalmente cambiada con relación a la versión anterior del programa.

Como introducción empezaremos a relatar algunos cambios que ha sufrido Strata en esta nueva versión. Lo primero que nos llama la atención son los altos requerimientos que necesita para poder funcionar, pues como mínimo necesitará 40 MB de RAM y se recomienda un micro procesador Power PC 604, si bien el autor lo ha instalado en un Power PC 601 y el funcionamiento es ideal. Lo primero que notaremos al abrir el programa es que el interfaz ha cambiado bastante, como podemos comprobar al abrir la ventana de texturas; se aprecia que está dividida en dos partes: la de arriba nos muestra las texturas que estamos realizando en nuestros modelos, y la de abajo las que tenemos en la librería. Asimismo, han cambiado los tipos de texturas y ahora existen *Solid, Surface* y *Volumetric*, además de haber introducido *Glows* y partículas.

El apartado de luces también ha cambiado bastante. Ahora existe la posibilidad de colocar luces volumétricas y *Lens Flares*, pudiéndose controlar mejor dónde influye una luz. Otro aspecto que también han sido modificado es cómo se aplican las texturas al objeto, así como dónde se fijan todas las transformaciones a un objeto determinado. Para terminar, señalar una función que resulta muy interesante, como es la posibilidad de introducir Plug-ins de terceros fabricantes (por ejemplo, el de esqueletos, que ya existe).

IMAGEN DEL NUEVO INTERFAZ DE STRATA.





TÉCNICAS AVANZADAS

3D MAX

Recreación de Atmósferas (II)
Autor: Christian Daniel Semczuk

Nivel: Avanzado
Plataforma: PC/MAC

En línea con el artículo anterior, este mes ofrecemos nuevos efectos aplicables al mismo ejemplo. Dichos toques harán de nuestro desarrollo una obra con mayor realismo y convicción.

Y es que se echaban en falta estos efectos. Detalles como un charco en el asfalto, algo inevitable en este tipo de situación atmosférica, y ya en animación, el fluctuar de la lluvia por efecto del aire, algunas gotas aisladas, consecuencia del acumulamiento de agua, bien sea sobre el árbol o el remate del muro, y alguna modificación más sobre el original.

Resumiendo el ejercicio del mes pasado, podríamos mencionar que fue el principio de lo que vamos a acometer en este nuevo episodio. Esta vez, hablaremos de él como si cuyo resultado final fuese a ser una animación lo más detallada posible, con la adición de los efectos que, en un principio, comentábamos. Retocaremos algunos de los pasos dados en el artículo anterior para dar aún más fuerza a la idea melancólica que deseábamos ofrecer con la animación.

MANOS A LA OBRA

Hecha esta pequeña introducción, podemos comenzar a trabajar. Y primeramente lo haremos con el charco, pero antes es recomendable prestar atención a una serie de especificaciones técnicas: *P90, 40 MB de RAM, Windows NT 4 y 3,2 Gb HD.*

Ante la frustración que produce la espera, tanto de *renders* de prueba como el refresco de pantalla, a la hora de redibujar el proyecto en nuestro ordenador (y no hablemos ya del render final, donde el ordenador puede estar días calculando los fotogramas) nuestra idea fue la de generar todo el proyecto en una máquina con las características que arriba mencionamos. De esta manera, pudimos comprobar que cualquiera puede seguirnos los pasos, tanto del artículo anterior como del que tenemos ahora entre manos. Lo que pretendemos con esto es animar a todos aquellos que no cuenten con el equipo necesario para trabajar profesional o semiprofesionalmente y, así, se atrevan a hacerlo con un proyecto de cierta envergadura.

EL CHARCO

A partir de ahora, de todos los detalles con que vayamos dotando a nuestra escena, el charco quizá sea el más complejo. Sobre todo, por el tiempo que se pierde con los *renders* de prueba y por su especial particularidad del material, que le dará las características necesarias para emular el agua.

Dicho esto, y antes de entrar en la creación de la geometría del charco, tenemos que preparar el camino, y nunca mejor dicho, pues debemos hacer una hendidura en el asfalto para que, de esta manera, el objeto charco, una vez creado, se superponga sobre la calle, dando una apariencia natural.

Para lograr dicho efecto, lo mejor en este caso es generar un mínimo de dos esferas, posteriormente achatarlas de manera irregular, y hacer una *Booleana* del asfalto a las esferas con carácter sustractivo. Conviene que el modelo asfalto tenga una geometría algo compleja, pues las operaciones *Booleanas* no suelen ser muy efectivas con poca resolución poligonal. Como dicho corte será la composición de varios modelos, es muy posible que el nivel de *Smooth* en las zonas donde interaccionen todos los objetos en el resultado final se aprecie de manera cortante. Para solucionar esto, y dar en esas zonas un suavizado para conseguir así una mejor visualización en el render, simplemente abrimos sobre el modelo un modificador *Edit Mesh*, y en el desplegable *Faces*, dentro de *Sub-Object*, vamos seleccionando todas las caras que han sido previamente cortadas y, a continuación, le damos un valor de 35 a 50 grados de *Auto Smooth.* En la figura 2 podemos observar tanto el resultado final de la *Booleana* sobre el "renovado" asfalto como la complejidad necesaria para dicho modelo.

Con el socavón en la calle, ya podemos generar el modelo del charco. Tenemos que tener en cuenta que la resolución o grado de detalle de éste ha de ser lo más grande posible. El porqué es sencillo: más adelante, después de situarlo y de aplicarle un material, le asociaremos un *Space Warp, Ripple* para ser exactos. Y es que este efecto, que simula las ondas producidas en el agua por la colisión de elementos, nos exige una geometría compleja para que, de esta forma, pueda efectuar su característica función de deformación de malla.

Para llegar hasta el modelo final, primero hemos de crear una caja o *Box*, de las *Standard Primitives*. Sin entrar en detalles,



FIGURA 1. ESCENA FINAL, CON LA ADICIÓN Y RETOQUE DEL EJERCICIO DEL MES PASADO.

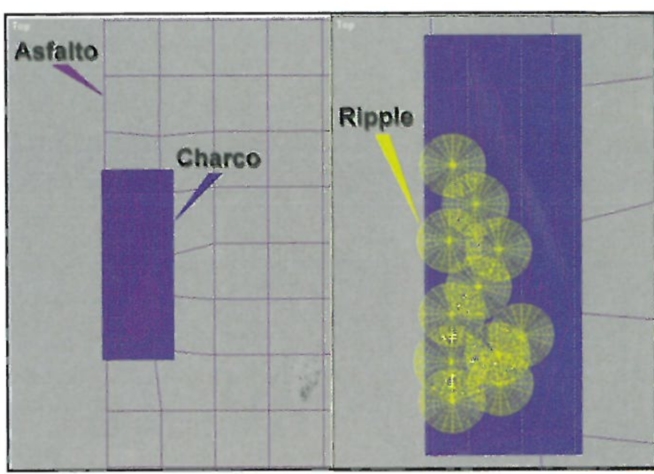


FIGURA 2. SITUACIÓN DEL CHARCO CON RESPECTO AL ASFALTO, Y DE LOS *RIPPLE*. OBSÉRVESE LA COMPLEJIDAD GEOMÉTRICA DE AMBOS MODELOS, UNO POR RAZONES DE SU POSTERIOR DEFORMACIÓN CON *SPACE WARP*, Y OTRO POR NECESIDAD, A LA HORA DE APLICARLE OPERACIONES *BOOLEANAS*.

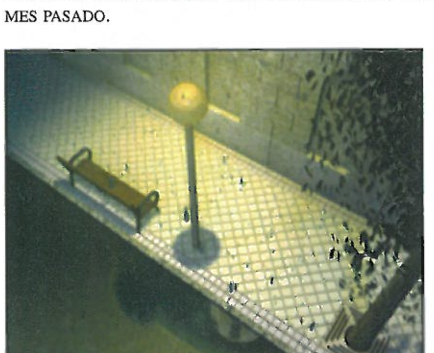


FIGURA 3. FOTOGRAMA DE LA SECUENCIA 3, EN LA CUAL SE PUEDE OBSERVAR LA CALIDAD FINAL DEL MATERIAL DE AGUA APLICADO AL CHARCO.

y ya sin más, la vamos presentando desde la vista *Top*. Cuando esté situada sobre su objetivo, le aumentamos los parámetros *Width* y *Length*, para que ajuste perfectamente a su sitio; luego le aumentamos las caras, con *Length Segs* y *Width Segs,* para conseguir un alto grado de detalle. Si observamos la figura 2 se aprecia la cantidad de caras con las que cuenta el modelo.

Hecho esto, se le aplica un modificador *Edit Mesh*, y ya en *Faces,* dentro de *Sub-Object* y seleccionando por caras, arrastramos el ratón desde una esquina inferior, abarcando así toda su extensión más baja, pero sin tocar las facetas que están por encima. Con la selección creada, sólo nos queda eliminarlas. El resultado final será un plano, con una gran complejidad, que estará perfectamente situado y con buenas expectativas para poder ser deformado por la acción del *Ripple*.

Llegados a este punto generaremos un material con el fin de emular el agua de un charco. Sus cualidades serán las siguientes: un marcado punto de brillo, incoloro por naturaleza; una reflexión del entorno bien pronunciada; su lógica transparencia y una propiedad adicional, sucio. Esta última cualidad sólo se verá en toda su plenitud desde un punto en el cual el ángulo de perspectiva sea elevado y esté a pie de cámara. De esta manera nos impedirá ver el fondo del charco (figura 3).

Estudiado el material desde un punto de vista meramente objetivo, entramos ya en el aspecto técnico: su creación digital desde el *Material Editor*. A partir de aquí nos remitimos al cuadro adjunto "Agua sintética", en este artículo, en el que obtendremos la información necesaria para afrontar este paso sin la necesidad de ir generando tantas pruebas como las que nos ha tocado hacer. ¡Y en semejante equipo!

Es hora de darle vida y dinamismo a nuestro pequeño charco, y lo haremos con los *Ripple*, situados en el panel *Create*, y dentro de los *Space Warps*. Seleccionado dicho efecto crearemos uno justo por debajo del modelo *Charco*, a fin de que actúe sobre la proyección ortogonal de la malla, deformándola conforme a la configuración del *Ripple*.

A continuación, ofreceremos los valores de configuración de estos característicos efectos de ondas. Hay que tener en cuenta que estos valores, fijados como aquí exponemos, puede que en el desarrollo del

AGUA SINTÉTICA

Ante la necesidad de emular el agua, tenemos que tener muy en cuenta sus características naturales. Nos ceñiremos, pues, a lo dicho en el apartado *El Charco*, con referencia al caso que ahora nos acomete.

Sin más preámbulos, y sin extendernos en tecnicismos, puesto que han sido revisados en el capítulo del mes pasado, entremos de lleno en los *Basic Parameters* del nuevo material:

- **Shading:** *Phong*.
- **Ambient:** *R:* 0, *G:* 0, *B:* 50
- **Diffuse:** *R:* 0, *G:* 0, *B:* 0. Pues es incoloro.
- **Specular.** *R:* 255, *G:* 255, *B:* 255
- **Filter:** *R:* 50, *G:* 50, *B:* 50. De esta manera lo oscurecemos, para contrarrestar la gran reflexión.
- **Opacity:** 20. Es un charco.

Los restantes parámetros no varían lo más mínimo con respecto al ejercicio del pasado mes.

En el desplegable *Maps* incluiremos una única opción, la reflexión. Así que, activando la caja situada al lado de la propiedad *Reflection*, escogemos la opción *Reflect/Refract*, y la configuramos de la siguiente manera:

- **Size:** 512. Recordemos que 3D Studio MAX 1.2 no trabaja con *Raytracing* selectivo. Entonces, para emular dicha propiedad, genera un mapa que posteriormente aplica sobre el objeto que tenga un material con esta característica. Así pues, cuanto más grande sea este mapa, más nitidez nos ofrecerá.
- **Blur:** Dejaremos activada esta opción, pero variándole el parámetro *Blur* a un valor de 0,2. La finalidad de lo aquí expuesto es conseguir que la reflexión del entorno no sea del todo borrosa. Cuanto más leve, mejor simularemos esta propiedad física.
- **Render:** Tal y como está por defecto.

Regresando al menú principal, fijaremos la cantidad de reflexión en torno al 75%. Ahora, y abriendo el desplegable *Extended Parameters*, el parámetro *Amt* lo llevaremos hasta un valor de 20 unidades. De esta manera dotaremos de una opacidad más uniforme a nuestro modelo.

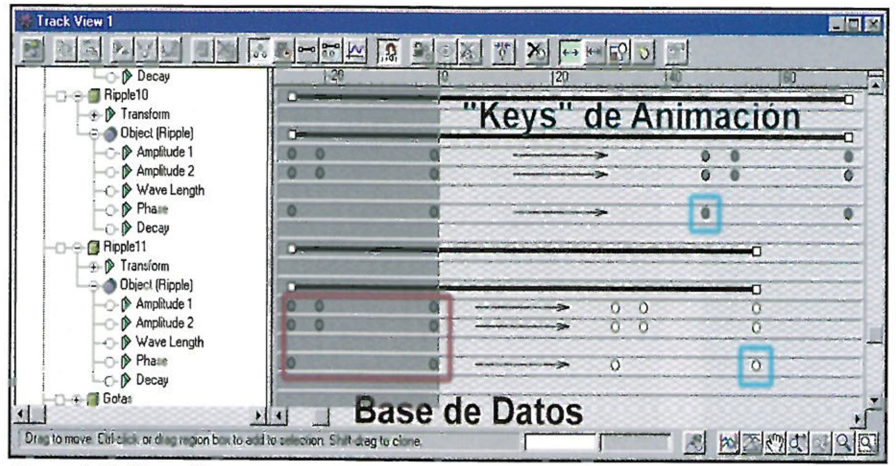


FIGURA 4. EL *TRACK VIEW*, INDISPENSABLE EN ESTE PROYECTO A LA HORA DE REPETIR DIFERENTES *KEYS*, A LO LARGO DE LAS PISTAS DE ANIMACIÓN.

lector no sean todo lo fieles que lo han sido para nosotros. Eso es debido, principalmente, a la extensión de la topografía del modelo al cual son aplicados y, sobre todo, al grado de detalle en la resolución geométrica. De todos modos no hay que desanimarse, pues una vez configurado de esta forma, bastará con cambiar algún parámetro un poco para que quedemos satisfechos con el resultado final.

Amplitude 1 y 2: *0 - 0,5.* Atención con estos valores. Ello nos da a entender que estando en 0 estas dos opciones, el grado de onda es nulo, mientras que en 0,5, pasan a ser el punto más álgido en el cual se produce la colisión de las gotas de lluvia con el agua. Su disposición en *Keys* de animación es la siguiente: 0= inicio, 0,5= cinco fotogramas más tarde, y vuelta a 0, veinte "frames" después. Estos parámetros determinan la altura de cada onda, y con estas variaciones hemos fijado las opciones para emular dicha colisión. Al principio, altas ondas y, posteriormente, éstas descienden de altura hasta confundirse con el plano del modelo del charco.

Wave Length: *1,5.* Esta configuración será fija en todas las secuencias. Aquí forzamos la longitud de onda. Como es lógico, para un charco de nuestra magnitud cuanto más pequeño, mejor.

Phase: *0 - 2,0.* Indica que comienza con un valor 0= inicio, y acaba veinticinco fotogramas después, con este parámetro 2,0. En este apartado indicamos la velocidad de expansión de las ondas.

Decay: *1,0.* Limitador del efecto de la onda. Si este valor fuese de 0, nuestra onda se expandiría a lo largo de todo el modelo donde esté aplicado el efecto. Con **1,0**, la regimos a un pequeño radio de acción.

Display: Con esta opción, si vamos alterando sus parámetros, sólo conseguiremos una manera de visualización, más grande o más pequeña, más detalle o menos, pero nunca variará su acción con respecto al modelo al que está aplicado.

Los parámetros animados deben ser activados todos al unísono

Una vez hecho todo esto, copiamos el efecto *Ripple* en la zona que deseemos de nuestro charco desde la vista *Top*. Ya sólo resta asociar cada uno de esos efectos al modelo charco, por medio de la herramienta *Bind to Space Warp*.

Los parámetros animados se activarán todos al unísono. El ir alternando dichos valores a lo largo de las pistas animación, es decir, que otorguen un cierto grado de caos y que, de esta manera, tengan un efecto más natural, lo haremos desde el *Track View*. En cuadro "Trabajar con el *Track View*" se explica cómo se puede hacer de esta tarea una correcta utilización, y ahorrarnos el consecuente gasto de tiempo que lleva el ir fotograma a fotograma creando *Keys* de animación.

ÁRBOL

En base a lo comentado el mes pasado, la técnica que utilizaremos esta vez será mucho más sencilla, ya que sólo emplearemos el modificador *Bend* para lograr el movimiento de las hojas, producido por la acción del viento. Como siempre, antes de proceder con este o cualquier otro efecto sobre quién sabe qué modelo, debemos pensar cómo vamos a afrontar dicho proceso, es decir, si es necesario modificar en algo el objeto. En el caso que nos acomete, tenemos un árbol compuesto por tronco, ramas y hojas, todo ello en un único modelo. Con nuestro guión en las manos, ya podemos observar que el movimiento será suave, pues el viento es más brisa que tal. Esto indica que sólo serán las hojas y las ramas más débiles las que se aprovecharán de la opción de animación sobre este modelo. El tronco, en este caso, permanecerá estático.

TRABAJAR CON EL *TRACK VIEW*

La manera más fácil y cómoda de simplificar procesos de animación, bien sea copiar *Keys* de animación como su posterior variación es, sin lugar a dudas, desde esta potente herramienta, el *Track View*.

Desgraciadamente, no vamos a poder extendernos en sus características y uso, pues por lo extenso del tema nos llevaría un capítulo entero, lo cual no significa que no podamos hablar de su uso especial para este proyecto.

Si tenemos que describir nuestro desarrollo, lo haríamos de la siguiente manera: sosegado y con ritmo. Y es que, generalmente, en toda la animación se aprecia más el ritmo marcado por la acción del viento y las ondas de colisión sobre el charco. Podemos decir que se trata de unas *Keys* fijas con las que, alternándolas, se obtiene una animación como la descrita anteriormente.

Ciñéndonos al ejemplo de los *Ripples*, lo desarrollaremos y veremos que esto puede ser aplicado a todos los objetos que necesiten del *Track View* para agilizar, de esta manera, algo que puede llegar a ser rutinario.

Habiendo creado las *Keys* sobre los *Space Warps*, abrimos ahora el *Track View*, buscamos los objetos *Ripple* y seleccionamos todas sus *Keys*. Hecho esto, las situamos por detrás del "frame" 0, es decir, dejamos su última *Key* en el fotograma -1.

Con esto hemos generado una pequeña "base de datos", a la cual recurrir para, posteriormente, copiarla a lo largo de las pistas de animación. Como nosotros queremos que el efecto de ondas sobre el charco se reproduzca de una manera un tanto caótica, si al *Ripple10* lo situamos desde su inicio en el "frame" 45, al *Ripple11* lo pondremos, por ejemplo, en el 31. La manera de conseguir esto es la siguiente: nos vamos a la "base de datos" del *Ripple10*, seleccionamos las *Keys* de animación de éste y, posteriormente, manteniendo presionada la tecla mayúsculas, arrastramos su primer *Key* hasta el "frame" 45. A continuación repetimos la misma operación con el Ripple11. Para tener un mayor conocimiento de este concepto podemos remitirnos a la figura 4 donde, de una manera ilustrada, hemos hecho lo mismo que lo descrito en este párrafo.

Esto es aplicable a todos los elementos que, por su animación, tengan un movimiento rítmico, con *Keys* comunes.

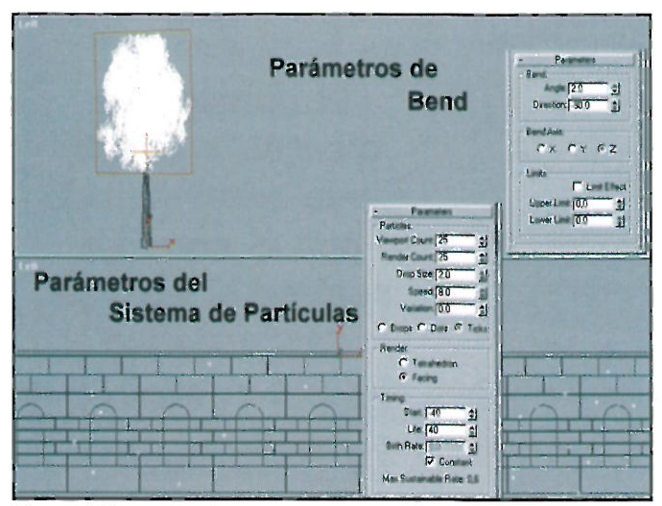


FIGURA 5. DEFORMACIÓN DE LAS HOJAS DEL ÁRBOL POR PARTE DEL MODIFICADOR *BEND*, Y SITUACIÓN DE LAS GOTAS SOBRE EL REMATE DEL MURO.

FIGURA 6. SITUACIÓN Y CONFIGURACIÓN DEL SISTEMA DE PARTÍCULAS Y EL *SPACE WARP, WIND*.

Así pues, dividiremos el modelo en dos objetos independientes. Para trabajar a gusto, esconderemos todo el entorno, bien sean modelos, luces o cámaras. A continuación, aplicamos a las hojas un modificador *Bend*, pinchamos sobre *Sub-Object* y elegimos *Center*. Lo que nos acontece ahora es arrastrar el *Center* hasta la base del modificador para que, de esta manera, la rotación se produzca desde este punto y no desde otro, dándonos como resultado final una apariencia irreal. En la figura 5 se obtiene la información necesaria para configurar a voluntad dicho modificador.

Hecho esto, generamos un par de *Keys* sobre el modificador, variando su ángulo de inclinación. Estas *Keys* serán nuestra "base de datos" para, posteriormente, repetirlas a lo largo de las pistas de animación (Ver cuadro "Trabajar con el *Track View*").

GOTAS AISLADAS

Para darle mayor detalle al proyecto en la animación, crearemos un sistema de partículas que emplazaremos a lo largo del remate del muro. Será estrecho y ocupará toda la longitud del mismo. Sus valores serán parecidos a los de la lluvia pero con la salvedad de que serán un poco más grandes las partículas, debido a que suelen ser gotas producidas por acumulamiento de agua. El *Render Count* lo limitaremos a sólo 25 partículas.

Para porcionar detalle emplazaremos un sistema de partícula a lo largo del muro

Por el tamaño de las gotas que producirá no le aplicaremos el efecto *Wind*, del cual explicaremos un poco más adelante, así pues tendrán una caída libre sin obstáculos u otro tipo de efecto adicional.

Dado que su configuración es bien simple, lo dejamos en las manos del lector. De todos modos, se puede observar una en la figura 5.

LLUVIA Y VIENTO

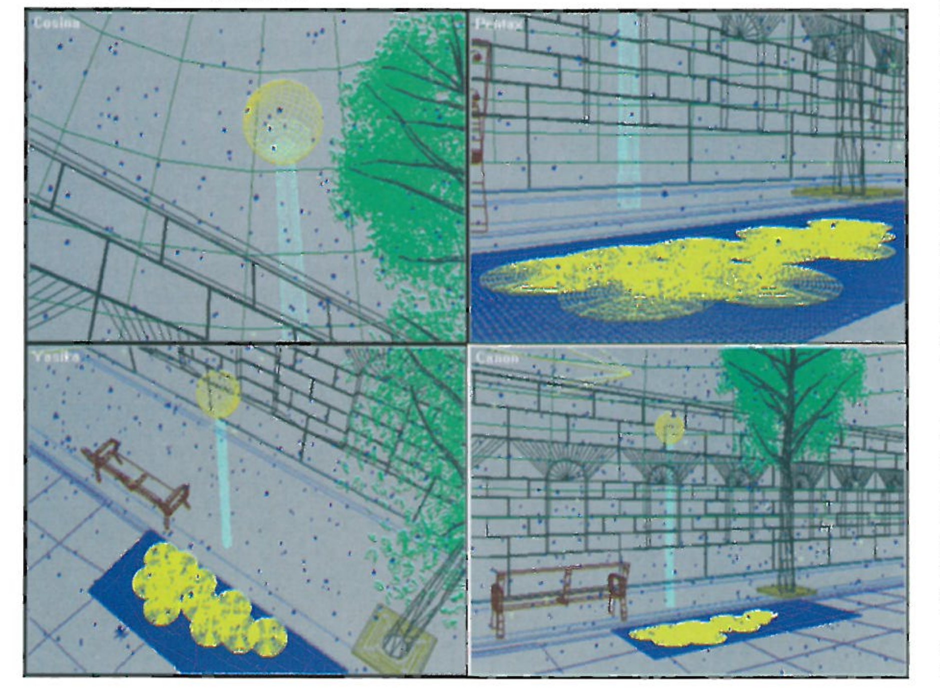
Ante un proyecto como éste, que basa su complejidad en materiales e iluminación más que en un gran despliegue geométrico, debemos, cuando menos, otorgarle una buena animación, máxime aún si se trata de la estrella de la película, en este caso la lluvia. Para que ésta cobre vida y nos arrastre dentro de la escena, debemos cuidar los detalles de su movimiento al máximo.

Pues bien, basándonos en lo expuesto anteriormente, tendremos que crear un efecto del que inevitablemente sólo se beneficiará este sistema de partículas. Éste es *Wind*, que lo podemos encontrar en el panel *Create*, dentro de los *Space Warps*.

Como se puede apreciar en la figura 6, sus valores de configuración son muy sencillos. Aquí sólo nos tomaremos la molestia de recordar que se trata de una leve brisa, así pues la opción *Strength* variará, en *Keys* de animación, de 0, para cuando el estado del aire sea nulo, 0,2, aumentando la fuerza del viento, y 0,5, en el momento más álgido de esta acción natural. Debemos ir alternando estos datos a un ritmo de tiempo de un segundo, es decir, cada 25 fotogramas un parámetro, otros 25, otro parámetro. Se hará más rápida la configuración si una vez hechas las *Keys* de animación, se guardan como base de datos y, a posteriori, son situadas a lo largo de las pistas de animación. Para entenderlo, podemos ver el cuadro "Trabajar con el *Track View*".

Otra cuestión a tener en cuenta también es que este aire tendrá efectos sobre el árbol ya que, luego, una vez renderizada nuestra animación, si no existe un compás de acompañamiento entre las gotas de la lluvia y el fluctuar de las hojas quedará de lo menos natural. Así pues, conviene que los valores de *Stregth* de *Wind* correspondan con los de *Angle*, del modificador *Bend* del árbol. Es decir, si con *Wind* nuestro valor es de 0,2, con

FIGURA 7. VIEWPORTS QUE REFLEJAN LA POSICIÓN DE LAS CÁMARAS EN LAS DIFERENTES SECUENCIAS, DENTRO DE NUESTRA ANIMACIÓN.



Bend éste será de unos 2 grados y con un ángulo de inclinación que se asemeje a la dirección en la cual corre el viento.

RESUMIENDO

Como se puede observar, se ha llevado a cabo una animación con cierto grado de complejidad, en un equipo no todo lo potente que desearíamos. Esto nos da pie a motivar a todos aquellos que no disponen de una estación de trabajo, que por sus características agilizan en gran parte cualquier proyecto que afrontemos.

Es conviene que los valores de *Stregth* de *Wind* correspondan con los de *Angle*

Resumiendo lo que os hemos ofrecido en estos dos últimos meses, llegamos a la conclusión de que, integrando en una escena poco volumen geométrico (léase modelos 3D) y habiéndolos estudiado previamente, creando sus materiales a conciencia, e iluminándolos acorde con un definido entorno, obtendremos unos resultados muy satisfactorios para nosotros mismos y para un hipotético espectador.

Sólo resta advertir de una cosa: que quien genera la escena, los modelos y todo en general, bien sea una ilustración o una animación, es el usuario y no el ordenador. Este último sólo es el medio para conseguir nuestra meta. A menudo, también se puede caer en la tentación de añadir un excesivo número de efectos especiales. Craso error. Generalmente, y si la idea a desarrollar es una historia cotidiana, como la que nos acontece, el número de efectos a añadir viene determinado por el entorno real en el cual se desenvuelve. Sin ir más lejos, una escena de ciencia ficción nos puede exigir cierto grado de complejidad y adición de diferentes tipos de efectos especiales. Ahora bien, si a esto le damos sólo toques sutiles, sin excedernos en dichos procesos, otorgaremos al espectador una idea que se creerá desde el primer fotograma, y sin cansarle la vista con tanto juego cromático.

El número de efectos a añadir viene determinado por el entorno real en el cual se desenvuelve

Para acabar, y volviendo a nuestra realización, mencionar la inclusión de una nueva secuencia producida por motivo de la adición del charco. La hemos situado entre la uno y la dos del proyecto comentado el mes anterior. En el CD-ROM adjunto a la revista la podremos observar y evaluar. Como venimos diciendo desde la pasada entrega, conviene practicar lo aquí expuesto y, posteriormente, comparar con nuestra película la animación realizada. Ésta es la única manera de agilizar la práctica con las herramientas del programa.



SERIE TÉCNICAS AVANZADAS

PHOTOSHOP

Un tremendo descuido

Autor: Julio Martín

Nivel: Medio
Plataforma: PC/MAC

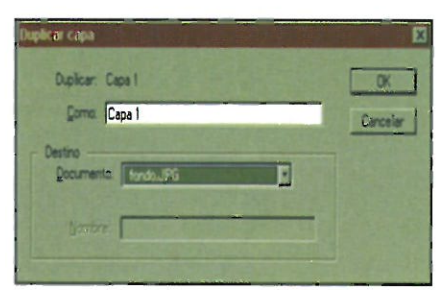
Nos hemos ido de vacaciones y hemos tirado un montón de fotos de monumentos, plazas... pero se nos olvidó hacer una de recuerdo con un retrato sobre un paisaje determinado. Tenemos el paisaje pero sin la persona delante. Vamos a intentar arreglar el descuido.

1 Se parte de 2 imágenes que escanearemos, una del paisaje y otra de la persona que queremos incluir. Si las dos fotos son del mismo carrete, o son del mismo viaje, con las mismas características de luz ambiente, etc, no tendremos muchos problemas. Pero para este ejemplo partimos de imágenes hechas en distintas fechas y con caraterísticas muy distintas.



2 En este caso la imágen del retrato tiene una luz muy baja y además proviene del lado opuesto al del paisaje.

Escaneamos las dos imágenes a tamaños similares, en este caso ambas a 300 puntos por pulgada, ya que luego la imprimiremos, y a un tamaño similar en altura. Alteramos la imagen del retrato "volteándola" horizontalmente, de modo que ahora sí viene la luz del mismo lado. Una vez hecho esto duplicamos la imágen con el retrato en una nueva capa del documento con el paisaje.



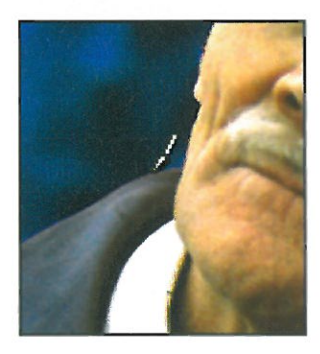
3 Una vez duplicada debemos colocar el retrato en una posición apropiada. En nuestro caso nos conviene poner el retrato en la parte inferior izquierda del paisaje porque tendríamos que reconstruir parte del brazo y porque en esa posición la foto queda más equilibrada.



4 Con la herramienta *Lazo* seleccionamos la silueta del retrato. Si pulsamos la tecla *ALT* mientras hacemos la selección ésta la realizaremos en líneas rectas, y así tendremos un mayor control si lo hacemos con pequeñas rectas. Si en un momento dado soltamos sin querer el ratón podemos añadir a la selección con la tecla *MAYÚSCULAS* y luego de nuevo con la tecla *ALT* para continuar con líneas rectas.



5 Ajustamos la selección al máximo con la opción *Máscara Rápida*. Seleccionamos un pincel pequeño y suave y un borrador con las mismas características y comenzamos a *"pintar"* sobre nuestra máscara.



6 Cuando tengamos la máscara perfectamente ajustada volvemos al modo de selección y borramos el fondo del retrato. El resultado debería ser como en el ejemplo. Si hemos utilizado un pincel demasiado pequeño antes de borrar el fondo del retrato deberíamos suavizar la selección. De este modo disimularemos más el recorte de la figura.



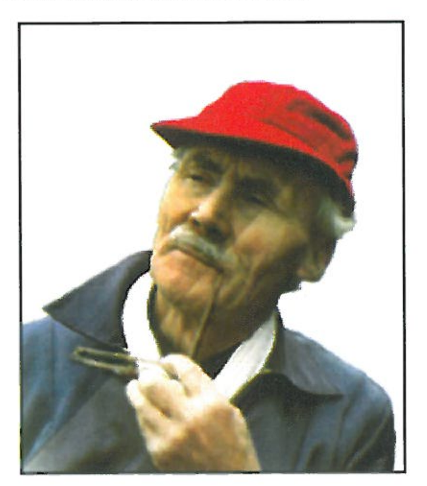
7 Ahora vamos a jugar con las luces tanto del fondo como del retrato para igualar un poco ambos, y alteraremos la profundidad de campo ligeramente. Primero duplicamos el fondo en una nueva capa y aplicamos sobre esta nueva capa un *Desenfoque Gaussiano* con un valor elevado, 15 en este caso.



8 Cambiamos el modo de esta nueva capa a *Normal* con un 52% de opacidad. Esto hace que el fondo se vea con un halo, pero aún nítido. El problema es que esto oscurece nuestro fondo. Podemos o bien aclarar la capa del fondo, o bien duplicar la capa que hemos desenfocado, y cambiar el modo de esta última a *Dividir* con un 75 % de opacidad. Ahora tenemos el fondo más claro y con un efecto similar al de un velo por delante de la cámara. En la imagen del ejemplo vemos, en la mitad superior izquierda, el efecto de la capa desenfocada al 52% en modo *Normal* sobre el fondo y en la parte inferior derecha el efecto que produce añadir la otra capa en modo *Dividir* al 75%.



9 Con el retrato vamos a hacer algo similar. Duplicamos la capa del retrato y aplicamos sobre esta un *Desenfoque Gaussiano* con un valor de 5. Cambiamos el modo de esta nueva capa a *Dividir* con una opacidad del 75%. Así, además de aclarar la foto, le damos un efecto similar que al fondo y no parecerá tan recortada. En el ejemplo vemos el efecto en la mitad inferior derecha.



10 El efecto final es como el de la foto siguiente, pero vamos a hacerle algo más al fondo.



11 Creamos una nueva capa entre las del fondo y las del retrato. Sobre ésta creamos un fondo degradado de negro a blanco radial, siendo el origen negro y estando éste en la parte inferior derecha del fondo. Lo que pretendemos es oscurecer muy ligeramente esa parte del fondo, de modo que cambiamos el modo de esta nueva capa a *Multiplicar* con una opacidad de 15%. De este modo los píxeles claros del degradado no afectarán la foto, pero los oscuros nos teñirán el paisaje ligeramente.



12 Ahora sólo tenemos que combinar las capas y ya tenemos el resultado que queríamos. Naturalmente, todo este proceso varía según sean las fotos de origen. Es importante elegir bien ambas para evitar tener que trabajar demasiado sobre ellas.

En este caso podemos además dar toques finales si, por ejemplo, antes de coombinar las capas aclaramos muy sutilmente algunas zonas del retrato con pinceles suaves y con muy poca opacidad, cambiando la luz que afecta al mismo, o alterando ligeramente el contraste del mismo.



Rebeca Allen, el cuerpo sintético

El trabajo de Rebecca Allen se centra, principalmente, en la representación sintética del cuerpo como forma de comunicación y expresión creativa, y utiliza la tecnología interactiva como una extensión del cuerpo y la mente.

El trabajo de la artista norteamericana, reconocido a nivel internacional, es de una gran versatilidad, ya que ha realizado obras en el ámbito de la animación por ordenador, las instalaciones interactivas y las performances multimedia.

Aunque su trabajo estuvo centrado, principalmente, en la animación por ordenador, en la actualidad está trabajando en un proyecto de arte interactivo llamado "The Bush Soul" y que fue presentado en la pasada edición de Art Futura.

Rebecca ha realizado varios trabajos de Animación por ordenador y performances en España, debido a la especial atracción que siente por nuestro país.

El cuerpo en movimiento

La artista se especializa en el estudio del movimiento y sus primeras obras se ocupan, concretamente, del movimiento humano

Entre los años 1980 y 1986 trabaja como artista e investigadora en el Laboratorio de Imagen por Ordenador del Instituto de Tecnología de Nueva York.



Videoinstalación.



Imagen de "Steady State".

Durante este periodo se centra en el estudio del movimiento y en sus formas de comunicación, ya que lo que más le atraía de un ordenador era la capacidad de crear movimiento, especialmente humano. En ese momento había muy pocos artistas interesados en ese campo, y todo era muy matemático y científico, así que decidió trabajar el movimiento humano para que las imágenes sintéticas no fuesen tan frías. Durante este periodo creó obras tan importantes como "The Catherine Wheel"(1982), una producción de videodanza de Twyla Tharp, y "Steps"(1982), un proyecto con el MIT y la Joffrey Ballet.

Rebecca se encargó de dirigir y diseñar en 1989 todo el material visual del álbum "Electric Cafe" del grupo alemán Kraftwerk, para el cual se crearon representaciones sintéticas de cada una de las cabezas del grupo. El trabajo también incluía la realización del vídeo de la canción "Musique non Stop", en el que se utilizaron programas especiales de animación facial.

En 1989 realiza "Steady State", una producción especial para el programa de TVE "El Arte del Video", donde hace un estudio del movimiento humano y de la relación entre los movimientos naturales y artificiales. Esta obra intenta reflejar las relaciones entre el hombre y la mujer a través del movimiento corporal de los dos protagonistas, y de cómo la mujer sueña hacia un mundo surrealista. Combina imagen real y sintética, reproduciendo los movimientos generados por el ordenador en sentido negativo, cubriendo los rostros humanos para centrarse en el movimiento.

A finales de los ochenta se centra en la vida artificial y en el movimiento de grupos, estudiando las interacciones entre elementos que se mueven juntos. Utilizando estas técnicas crea en vídeo de alta definición, "Behave" utilizando el programa de comportamiento de manadas de Craig Reynolds.

España

La primera vez que llegó la artista a España fue en 1987 para presentar una conferencia en Barcelona y se enamoró del país a primera vista, por el espíritu de la gente, del ambiente, de sus colores y de las texturas de sus ciudades. En 1989 regresa para producir en Animática la obra "Steady State" para TVE.

A finales de 1990 viaja a Barcelona para colaborar con

la compañía teatral La Fura dels Baus para la creación de la pieza *"Mugra"* con 12 actores y 70 monitores, comisionado por Art Futura. *"Mugra"* es una performance que integra música, interpretación y elementos visuales unidos por un hilo argumental.

En 1991 contribuye al "Palacio de la Memoria" con una performance multimedia con actores, tres pantallas panorámicas de cine, y música de John Paul Jones y Peter Gabriel. Para la creación de las imágenes utilizó un simulador de vuelo de Evans & Sutherland en Rediffusion Simulation, Ltd. (Inglaterra). Este mismo año el Pabellón de España encarga a Rebecca la dirección de dos piezas, "Fuego y Aire" y "Agua y Tierra", para representar a España en la Expo´92 de Sevilla. Para su creación, se reunieron en el Museo Reina Sofía para averiguar cuáles eran los símbolos representativos de España. En la obra "Fuego y Aire" intenta reflejar la idea de energía y de fuerte vitalidad de España a través de estos dos símbolos. La música fue compuesta por Juan Arteche según se diseñaba la animación, de ahí que exista una fuerte conexión entre la música y las imágenes. Estas dos obras fueron seleccionadas para el Electronic Theatre de Siggraph´92. Un año mas tarde crea en Barcelona "Laberint" un corto encargado por TV3 y Animática.

Arte interactivo

El primer contacto que tuvo con la tecnología interactiva y la realidad virtual fue en el MIT, a finales de los 70, donde trabajó en el proyecto "The Aspen Movie Map" que utilizaba el primer disco láser conectado a un ordenador. Rebecca Allen estaba muy interesada en el arte interactivo, pero los ordenadores eran muy lentos, y sólo se podían crear formas y colores muy simples, así que decidió acudir al Laboratorio de Imagen por Ordenador del NYIT y centrarse en la animación 3D que daba mejor calidad de imagen.

Según Rebecca, a finales de los ochenta todo el mundo hablaba de *realidad virtual*, pero los mundos no eran muy interesantes, porque los ordenadores no eran lo suficientemente potentes como para conseguir imágenes fluidas, a no ser que se trabajase con costosos simuladores de vuelo.



Imagen tridimensional generada en tiempo real.

Entre los años 1992 y 1993, la tecnología era bastante rápida y barata, y el sector de los videojuegos comenzó a centrarse en mundos tridimensionales interactivos. Durante su estancia en Virgin pudo aprender gran parte de esta tecnología, pero decidió abandonar la empresa porque no podía aplicar los nuevos conocimientos aplicados al arte interactivo, en una compañía comercial.



Imagen de "The Bush Soul".

Después de este periodo regresó a la universidad de UCLA, para experimentar con *tiempo real*, y trabajar con equipos cedidos por Intel Corporation y un gran equipo de estudiantes de diseño y ciencias, para crear un programa llamado "Emergence". Utilizando este programa han desarrollado la obra "The Bush Soul". Rebecca intenta dar vida a la *realidad virtual*, centrándose en el comportamiento de los personajes y de los objetos del mundo virtual, para que tengan vida y respondan al ambiente social cuando la gente entra. También investiga las relaciones entre el *avatar*, representación virtual de nosotros mismos en forma de personajes generados por ordenador, y la persona, examinando la cantidad de control del artista y el participante.

Para Rebecca el arte interactivo tridimensional es un área muy importante para enfocar su trabajo y este año con su equipo creará una pieza para presentar en la próxima edición de Siggraph.

Juan Carlos Olmos 3D

Su última creación fue presentada en la pasada edición de Art Futura. Actualmente está trabajando en arte interactivo tridimensional.

BIOGRAFÍA

Rebecca Allen es una artista multimedia reconocida internacionalmente, que utiliza la más avanzada tecnología para sus creaciones.

Licenciada por la Rhode Island School of Design en 1975, obtuvo el Master en el Massachusetts Institute of Technology en 1980. Fue miembro del Architecture Machine Group del MIT, participando en proyectos de videodiscos interactivos.

Entre los años 1980 y 1986 fue directora, diseñadora e investigadora en el Computer Graphics Lab del New York Institute of Technology, creando obras como "Steps" o "The Catherine Wheel".

Antes de entrar en Virgin Entertaiment en 1993 realizó numerosos encargos en Europa y Estados Unidos. En Virgin ejerció de directora creativa y visionaria 3D investigando las posibilidades creativas de la tecnología interactiva de los juegos.

En la actualidad está creando una instalación interactiva llamada "The Bush Soul", con elementos de inteligencia artificial, redes y espacios virtuales tridimensionales. Rebecca preside el departamento de diseño y es codirectora del Center for the Digital Arts en la Universidad de California, Los Angeles (UCLA).





CURSO SOFTIMAGE

Las curvas de función en la animación

Autor: Juan Carlos Olmos

Nivel: Medio

Una de las partes más interesantes y completas de Softimage 3D es el módulo de animación, que permite recrear cualquier tipo de movimiento por complejo que sea gracias a sus avanzadas y potentes herramientas.

En el número de este mes se hará una pequeña introducción a la animación en Softimage a través del modulo *Motion* (Movimiento), aparte de ver cómo funciona la ventana *fcurve* (curvas de función, que permiten ajustar o modificar una animación). También se realizará un pequeño ejemplo de animación de una pelota botando para ver, de forma práctica, cómo utilizar estas ventanas y algunos comandos de animación.

Las *Fcurve* representan de forma gráfica la relación entre el tiempo y el valor del factor animado, que puede ser la rotación, la traslación o cualquier otro parámetro que se pueda animar. Permiten cambiar las posiciones de los *keys* de animación, el tipo de interpolación, la continuidad e inclinación de la curva. Asimismo, las curvas se pueden cortar, copiar o pegar en su totalidad o en segmentos específicos.

La ventana *DopeSheet* permite crear cambios de una forma más global modificando los *keyframes* representados por rectángulos alargados, con herramientas tradicionales de cortar, copiar y pegar.



SELECCIÓN DE CURVAS

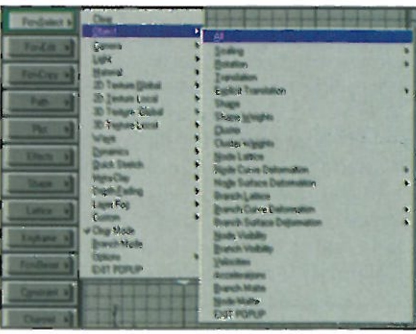
Para seleccionar las curvas de un objeto animado y que aparezcan en una ventana se debe acceder al comando *FcrvSelect/Object/* (figura 1), que se encuentra en el módulo *Motion,* y elegir el parámetro que ha sido previamente animado. Si se quieren ver las curvas de varios objetos distintos a la vez se seleccionarán con *Multi* y luego se accederá al comando de selección de curvas(figura 2) para seleccionarlas.

Por ejemplo, si se ha animado la traslación de un objeto y se quieren seleccionar sus curvas de función, se debe ejecutar el comando *FcrvSelect/Object/Translation,* o si es el *Roll* de la cámara *FcrvSelect/Object/Camera/Roll.*

Una vez que han aparecido las curvas del objeto en la ventana, se puede seleccionar una curva pulsando la barra espaciadora y pinchando sobre su nombre o directamente sobre ella. Otra posibilidad es seleccionar varias curvas teniendo la opción de selección *Multi* activada.

Para borrar las curvas de la ventana *Fcurve* se deberá activar el comando *FcrvSelect/Clear.* Cuando está activado el comando *FcrvSelect/Clear Mode,* al seleccionar una nueva curva borra las anteriores

FIGURA 1. MENÚ DE SELECCIÓN DE CURVAS.



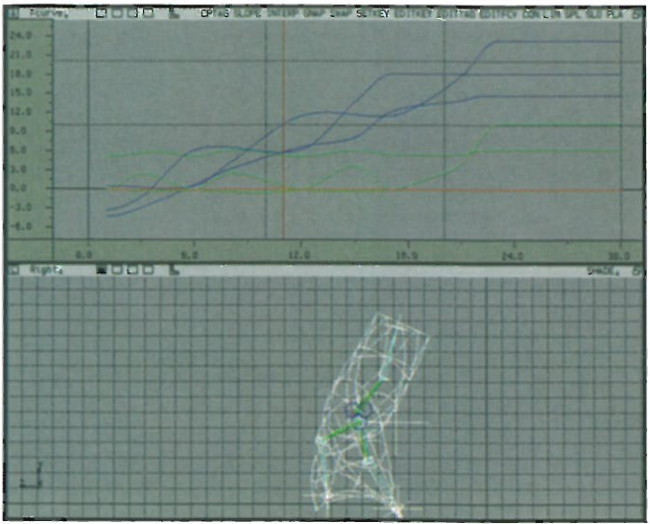


FIGURA 2. SELECCIÓN MÚLTIPLE DE CURVAS.

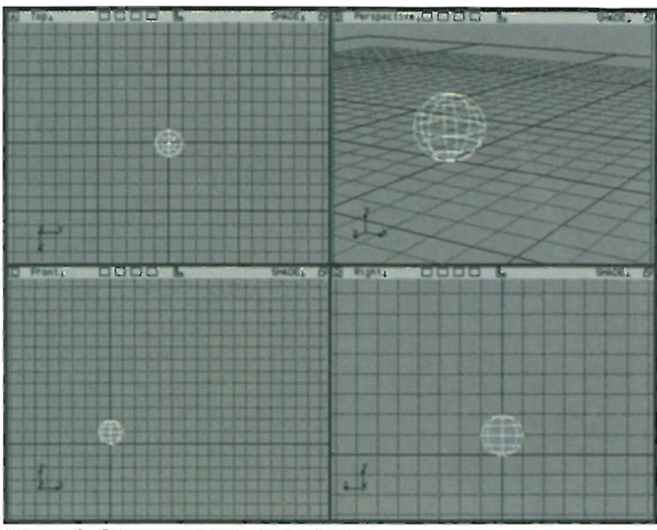


FIGURA 3. OBJETOS DEL EJEMPLO PRÁCTICO.

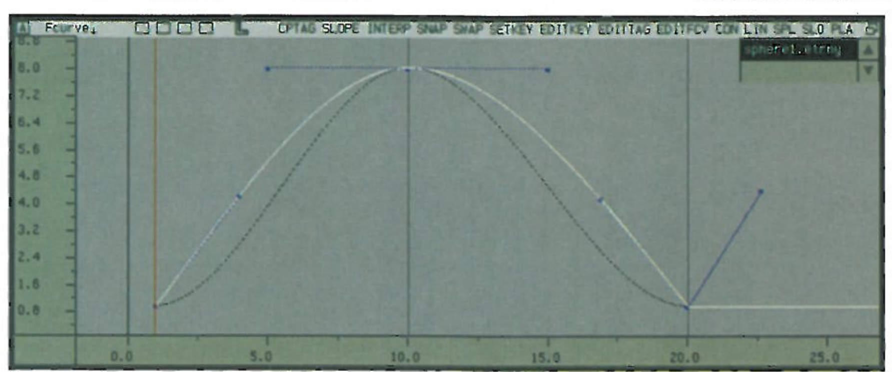


FIGURA 4. REPRESENTACIÓN GRÁFICA DEL BOTE DE LA PELOTA.

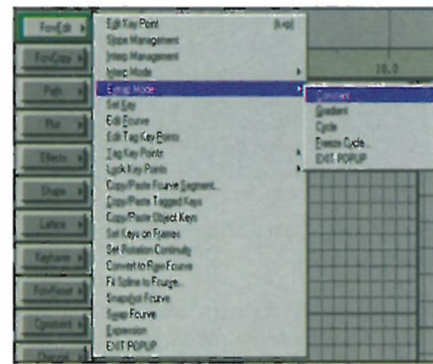


FIGURA 5. MENÚ DE EDICIÓN DE CURVAS.

de la ventana. Sin embargo, cuando está activado las va acumulando.

LA VENTANA *FCURVE*

En esta ventana se pueden modificar las curvas de función o cambiar la animación. Los puntos rectangulares azules representan los *Keyframes*, se les llama *key points*, y sólo aparecen cuando está seleccionada la curva. La interpolación entre los puntos se representa por una línea que los conecta, mientras que el tiempo está representado en el eje horizontal y el valor animado en el vertical. Los nombres de las curvas aparecen en un rectángulo en la parte superior derecha de la ventana, y su color corresponde con el de la curva que representa.

Para poder desplazarse dentro de la ventana *fcurve* se utiliza la conocida opción *Zoom* que se ejecuta con el comando *Camera/Zoom,* o bien pulsando la letra *z*. Con el botón izquierdo del ratón se desplaza la vista, y con el central y el derecho se acerca o se aleja respectivamente. Si se pulsan simultáneamente las teclas *z* y *x*, el desplazamiento de la vista y el zoom se realizarán exclusivamente en el eje horizontal, y con las teclas *z* y *s* en el vertical.

En la parte superior de la ventana se puede ver una barra con una serie de opciones que permiten editar las curvas. Dependiendo de la resolución y del tamaño de la ventana, se verán mas o menos opciones. Los comandos de la barra son los siguientes:

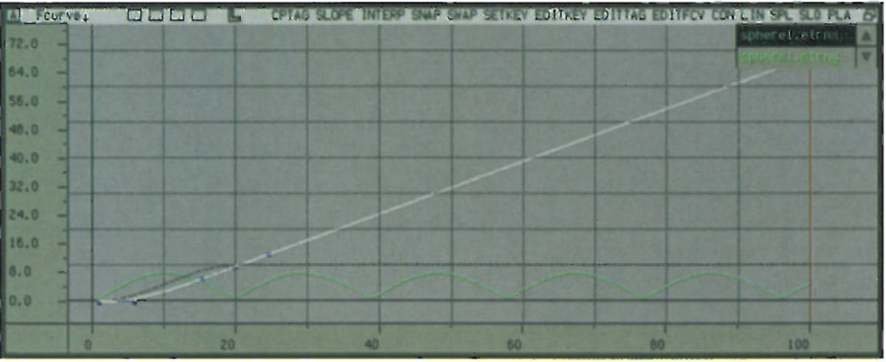


FIGURA 6. CURVAS DE MOVIMIENTO EXTRAPOLADAS.

• *CPTAG:* Permite cortar, copiar o pegar *keys* que estén seleccionados en modo *TAG*. Pulsando el botón derecho se copian en un *buffer,* con el central se borran y guardan en el *buffer* y con el derecho se pegan sobre el punto de la curva que se desee.

• *SLOPE:* Este comando se utiliza para ajustar la continuidad e inclinación de una curva *Spline* en un punto o *key.* Con el botón derecho del ratón se unifican las inclinaciones, con el central se rompe la continuidad y con el derecho se recupera la continuidad inicial.

• *INTERP:* Permite cambiar el tipo de interpolación de cada segmento de una curva, y pueden ser *Spline, Linear* y *Constant.* Dependiendo del botón del ratón que se pulse se aplicará un tipo de interpolación u otro.

• *SNAP:* Este comando realiza una copia de la curva en el fondo de la ventana, representada por líneas discontinuas. Esto permite modificar la curva teniendo como referencia la curva del fondo.

• *SWAP:* Cambia la curva actual por la de líneas discontinuas. Este comando es útil para realizar pruebas con las curvas ya que, en cualquier momento, se puede regresar a la original.

• *SETKEY:* Permite editar una curva de forma numérica a través de una ventana de edición.

• *EDITKEY:* Este comando se utiliza para editar una curva. Si se pulsa con el botón izquierdo del ratón se podrá cambiar la posición de un *key* o cambiar los segmentos de inclinación de las curvas. Con el botón central se insertan más *keys* y con el derecho se eliminan.

• *EDITTAG:* Se utiliza para modificar puntos seleccionados en modo *TAG*. El botón izquierdo permite trasladar los pun-

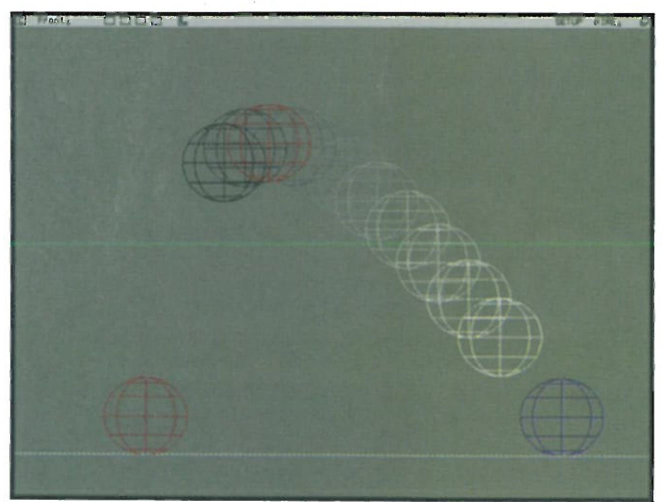



FIGURA 7. MOVIMIENTO DE LA PELOTA EN MODO *GHOST*.

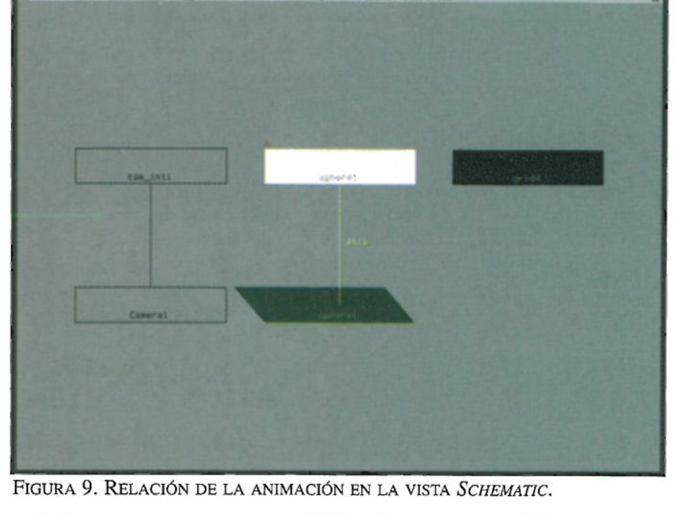


FIGURA 9. RELACIÓN DE LA ANIMACIÓN EN LA VISTA *SCHEMATIC*.

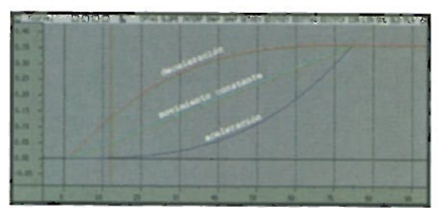


FIGURA 8. DISTINTOS TIPOS DE MOVIMIENTO.

tos, el central escalarlos en el eje *X* y el derecho en el *Y*.

• *EDITFCV:* Es muy similar al comando anterior, la única diferencia es que aplica las modificaciones a toda la curva y no a los puntos seleccionados.

Las siguientes opciones determinan el tipo de interpolación global entre los distintos *keys*:

• *CON*: Repite el valor del *key* hasta que llega al siguiente, produciendo movimientos bruscos y con saltos.

• LIN: Crea intervalos regulares entre los *keys*, uniéndolos con líneas rectas.

• SPL: Con esta opción se crean movimientos suaves, con aceleraciones y deceleraciones entre los *keys*, uniéndolos con curvas *splines*. Es el tipo de interpolación por defecto.

• SLO: Es similar al anterior, pero mantiene la inclinación de los modificadores horizontalmente.

• PLA: Cambia el *slope* o inclinación de una curva a 0 si el siguiente *key* tiene el mismo valor, o es un máximo o un mínimo.

EJEMPLO PRÁCTICO

Para ver cómo se utilizan las *Fcurves*, se va a realizar un ejemplo práctico y sencillo de una pelota botando. La modificación de las *Fcurves* permitirá crear el movimiento en muy pocos pasos, con muy pocos *keys*, y editarlo de forma muy sencilla.

El primer paso consistirá en acceder al módulo *Model* para crear los modelos de la animación (en este caso una pelota y un suelo). Para crear la pelota se utilizará el comando *Get Primitive/Sphere* con las opciones *NURBS* y *Cubic* activadas, y con un tamaño de la esfera de radio 1. Para el suelo sobre el que bota la pelota se accederá al comando *Get/Primitive/Grid* introduciendo un valor de 10 al tamaño del las celdas en los ejes *x* y *z*, y 5 en el número de celdas. A continuación se desplazará la pelota en el eje *y* para colocarla sobre el suelo (figura 3).

Una vez creados y colocados los objetos, se accede al módulo *Motion* para crear los *keys* de movimiento de la pelota. Primero, se seleccionará la esfera en el fotograma 0 y se pulsará el comando *Save/Object/Explicit Translation/Y*, para grabar su posición en el eje y en ese fotograma. A continuación, se deslizará la barra de tiempo hasta el fotograma 10, donde se trasladará la pelota ocho unidades en el eje *Y* y se grabará su *key* de posición con el comando anterior. Una vez realizado esto, se tendrá que grabar la posición de la esfera en el fotograma 20 para que realice el bote completo. La forma más sencilla de hacerlo es ir al fotograma 0 y desplazar la barra de tiempo con el botón derecho, de forma que avancen los fotogramas pero los objetos no se animen, y una vez situado el cursor en el fotograma 20, grabar el *key* de translación con *SaveKey/Object/ Explicit Translation/Y*. Si se pulsa el botón *Play* se verá cómo la pelota realiza un bote completo del fotograma 0 al 20.

Cuando se tienen los *keys* de la animación grabados es necesario ajustar las *fcurves* para conseguir el efecto deseado. Al visualizar la animación se podrá ver cómo la pelota decelera al llegar al suelo, y eso no debería ocurrir. Para solucionarlo se activará la ventana *Fcurve* en una de las ventanas superiores y se accederá al comando *FcrvSelect/Object/Explicit Translation/Y*, de forma que aparezca la representación gráfica sobre el tiempo del movimiento de la pelota en el eje Y. Una vez representada la curva en la ventana, se tendrá que ajustar su inclinación en el punto de entrada y de salida con el comando *EDITKEY* y pulsando el botón izquierdo, de forma que queden lo más rectas posible (con una inclinación aproximada de unos 45 grados), para evitar la aceleración inicial y la deceleración final (figura 4).

Cuando el movimiento de un bote de la pelota está ajustado y es el deseado, se puede repetir a lo largo de la animación sin tener que crear nuevas *keys*. Para ello, se creará un ciclo de la animación, es decir, que el movimiento en el eje *Y* se repita indefinidamente. El ciclo se creará seleccionando la curva y accediendo al comando *FcrvEdit/Extrap Mode/Cycle* (figura 5), creando de forma automática el movimiento de repetición (figura 6). Si se pulsa *Play* se verá cómo la pelota bota desde el primer fotograma hasta el último. Si se desea, se puede ajustar de nuevo la curva para cambiar la altura o duración del bote, simplemente modificando la curva original, viendo cómo el cambio afecta al resto de la curva.

Para conseguir que la pelota bote avanzando en el eje *X*, se deberá ir al fotograma 0, grabar un *key* de translación con el comando *SaveKey/Object/Explicit Translation/X*, desplazar la pelota 10 unidades en el eje *X* y grabar otro *key*. Si se pulsa *Play* se verá cómo la pelota avanza hasta el fotograma 20, pero luego no continúa. Para que avance durante toda la animación, se deberá extrapolar la curva con el modo *Gradient*. Para conseguirlo se seleccionará la curva de traslación en el eje *X*, con el comando *FcrvSelect/Object/Explicit Translation/Y* y se activará el comando *FcrvEdit/Extrap Mode/Gradient*. A continuación, se modificará la inclinación en el segundo *key* de la curva, para que el movimiento sea continuo en X (figura 6). Si se pulsa *Play* se podrá ver cómo bota y avanza la pelota a lo largo de toda la animación.

Una vez definidos los movimientos en los dos ejes, se puede ajustar con total libertad el movimiento de la pelota variando los *keys* iniciales. Si se quiere liberar al movimiento en *Y* del ciclo, se deberá activar el comando *FcrvEdit/Extrap Mode/Gradient/Freeze Cycle*.

1er concurso 3D World de Imágenes y Animaciones

Ganadores del sorteo Manual Técnico 3D Studio MAX

MIGUEL ANGEL DEL POTRO MARTIN
C/ Mercurio Nº 6
28032 MADRID

ALBERTO SENDIN
C/ Valdeciprián Nº 25
Daganzo
28841 MADRID

ENRIQUE MONFORT VALERO
C/ Iberia Nº 12
Hospitalet de Llobregat
08906 BARCELONA

MIGUEL ANGEL RUIZ BURRUESO
C/ Bofarull 38
08027 BARCELONA

JAVIER ROLLON MORAN
C/ Ferraz Nº 8
MADRID

FCO. JAVIER FERNANDEZ NIETO
C/ Ancora Nº 12
28045 MADRID

JODE RAMON GARCIA
C/ Valladolid Nº 22
Alcorcón
28922 MADRID

BORJA MORALES HERNANDO
Avda de Navarra Nº 18
20013, SAN SEBASTIAN

JESUS LANZA SALCINES
C/ Sta. Teresa Nº 32
Maliaño
39600 CANTABRIA

JOSE ERNESTO CHICA DE LA TORRE
C/ Sierra Magina Nº 21
23008 JAEN

ANDRES VIZUETE MARIN
Avda. La Fama Nº 17
Edif. Rodas
30006 MURCIA

MANUEL PARRA LOZANO
C/ Doctor Manuel Villegas Nº 4,
14007 CORDOBA

ANTONIO JAVIER SAIZ NUÑEZ
C/ Agustin Gonsalvez Orts BL 4, P 3
03010 ALICANTE

GONZALO BLAZQUEZ GUTIERREZ
Avda. Casaquemada Nº 18
El Plantío
28023 MADRID

DANIEL SMALLMAN
C/ Altos de Capuchinos Nº 13
Antequera
29200 MALAGA

FRANCISCO ALONSO DIAZ
Avda. Fuerzas Armadas Nº 29
Getafe
28901 MADRID

ANGEL ESPINOSA BEATO
Avda. Costa Blanca nº 84,
Bloque III
San Juan Playa
03540 ALICANTE

CARLOS GUTIERREZ MINTEGUI
C/ Montes y Martin Baro Nº 11
47008 VALLADOLID

JUAN GARRAZA ZURBANO
Aptdo. Correos 4048
31080 PAMPLONA

PEDRO JAEN RODRIGUEZ
C/ San Francisco Nº 19
Lucena
14900 CORDOBA

DANIEL TOMAS BOSCH
C/Almansa 85-7
08042 BARCELONA

FERNANDO VICEN CRESPO
C/ La Coruña 34-36-38
ZARAGOZA

PAULO BARBOSA
Rua Augusto Costa
(Costinha) Nº 23
1500 LISBOA (PORTUGAL)

ENRIQUE CERREJON TOCA
C/ Aldeanueva de la Vera
Nº 26
28044 MADRID

PEDRO RÓDENAS TORRECILLAS
C/ León Fontova Nº 23
Badalona
08913 BARCELONA

ANTONIO ESPAÑA CARDENAS
C/ Mateo Alemán 2
21005 HUELVA

PEDRO BIXQUERT PEDRO
Camino Viejo de Valencia Nº 112
Grao de Gandía
46730 VALENCIA

JOSÉ MEDINA
C/ Llauder 123-125
Mataró
08302 BARCELONA

JUAN RAMON AMOROS PRIETO
C/ Felipe II, 10
Novelda - 03660 ALICANTE

LUIS RUIZ CRESPO
Aptdo. Correos 213
Alcorcón
28922 MADRID

JUAN J. MALPARTIDA
Calle Ancha Nº 19
11001 CADIZ

ALBERTO CORRAL
C/ Diderot 23
35007 LAS PALMAS DE GRAN CANARIA

LUIS ANGEL MENDAÑA DEL RIO
C/ Conde de Toreno Nº 18
24006 LEON

JORGE BLANCO MALDONADO
Avda. País Valenciá 65
Torrent
46900 VALENCIA

RODRIGO BLAAS
C/ Las Flores Nº 4
18004 GRANADA

CARLOS DOMENE
C/ Encomienda de Palacios Nº 18
28030 MADRID

LLUIS MARTINEZ HUGUET
C/ Asturias Nº 70
08024 BARCELONA

FRANCISCO PARDO MARTIN
C/ Palleter Nº 44
Paterna - 46980 VALENCIA

JORDI SANS ALSINA
Avda. Josep Tarradellas Nº 170
L'Hospitalet de Llobregat
08901 BARCELONA

JUAN REINA SILVA
Avda. Reina Mercedes Nº 19-C
41012 SEVILLA

MANUEL RICO FREIRE
C/ Dionisio Alcalá Galiano Nº 15
41009 SEVILLA

JOSÉ MANUEL PEREZ DE VRIES
C/ México Nº 39
Coslada
28820 MADRID

CARLOS PRADO MOTA
C/ Villa de Plencia Nº 4
Las Arenas
48930 VIZCAYA

PEDRO ANTONIO GOMEZ GOMEZ
C/ Constitución Nº 6
19003 GUADALAJARA

FRANCISCO CUENCA ALCARAZ
C/ Real Nº 65
Torredonjimeno
23650 JAÉN

VERONICA ESPERICUETA BLAZQUEZ
C/ Alcorisa Nº 77
28043 MADRID

LUIS ROJO CALLEJO
C/ Carlos III Nº 10
28022 MADRID

DAVID RIVERA DUQUE
V/ Francisco Pizarro Nº 32
Getafe
28903 MADRID

ENRIQUE GATEM BREIS
C/ Sant Antoni Maria Claret Nº 8
08037 BARCELONA

FRANCISCO GOMEZ SANCHEZ
C/ San Martín Nº 91, 2º P11
38004 STA. CRUZ DE TENERIFE

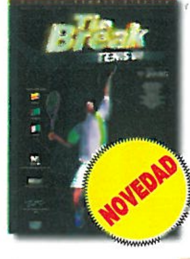
VICTOR D. LOPEZ CERCAS
C/ La Nacho Nº 9
Alcorcón - MADRID

• Consiga ya su libro, videojuego o multimedia favorito • Haga su pedido llamando al (91) 304 06 22 •



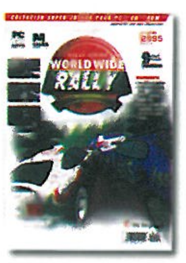
Contiene ocho de los mejores juegos de todos los géneros existentes: Arcade, Aventura, Deportivo, Simulador... Una colección imprescindible repartida en 2 CD-Rom.

2.995 ptas.
Incluye 2 CD-ROM.



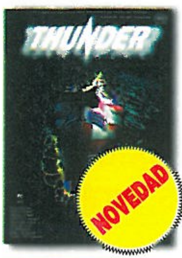
El juego de tenis más espectacular. El simulador 3D definitivo de tenis. Con increíbles efectos visuales como destellos de sol, sombras dinámicas, fuentes de luz y cámaras móviles que seguirán la acción en todo momento.

2.995 ptas.
Incluye CD-ROM.



Un arcade de carreras en 3D que hará las delicias de todos los *fitipaldis*. Gracias a su alta resolución, a su soporte multijugador y a sus 15 cuidadas pistas podrás sentirte un auténtico piloto al volante de tu coche preferido.

2.995 ptas.
Incluye CD-ROM.



Modelos 3D voxelizados de geometrías superiores a 50 mil polígonos, paisajes fotorrealistas con resolución automática y superficies reflectantes gracias a la técnica VTP. 3 Equipos, 6 thunder Arrows y 15 circuitos diferentes.

2.995 ptas.
Incluye CD-ROM.



Permite crear juegos comerciales y libres de royalties. Posee un entorno integrado que incluye un diseñador gráfico, generador de fuentes de letras y explosiones, así como 15 juegos y multitud de tutoriales.

4.995 ptas.
Incluye CD-ROM.



Con variadas novedades y mejoras entre ellas, Iberconta 2.0 con la incorporación del módulo de profesionales (Estimación objetiva y Directa) e Iberfacturación 2.0, que permite la impresión de facturas sin límite de páginas.

2.995 ptas.
Incluye CD-ROM.



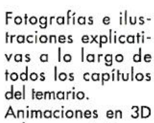
SuperLoto 1.0 es una herramienta pensada para los aficionados a los juegos del 6-49, Lotería Primitiva, Bonoloto y El Gordo de la Primitiva. Es un programa informático para uso doméstico, oficinas de apuestas y peñas.

1.995 ptas.
Incluye CD-ROM.



Programa que ayuda al usuario a llevar un seguimiento regular de las jornadas para poder realizar quinielas con el mayor acierto posible. Una herramienta muy útil para los aficionados al fútbol.

1.995 ptas.
Incluye CD-ROM.



Fotografías e ilustraciones explicativas a lo largo de todos los capítulos del temario. Animaciones en 3D sobre mecánica y diversas maniobras con vehículos.

2.495 ptas.
Incluye CD-ROM.

Incluye preguntas de las oposiciones más importantes del Estado. Entre ellas se encuentra el MIR, Administración del Estado, Auxiliar Administrativo, Administración del Inem.

2.495 ptas.
Incluye CD-ROM.



Contiene base de datos con los directores que han intervenido en las producciones cinematográficas de 1996 y una descripción detallada de las mismas.

2.495 ptas.
Incluye CD-ROM.



Viaje temático sobre la historia y la filosofía. Incluye los últimos temarios y cuestionarios de filosofía para selectividad.

2.995 ptas.
Incluye CD-ROM.

Esta publicación electrónica tiene como objetivo dar a conocer los motivos, acontecimientos y consecuencias de la Guerra Civil Española vivida en los años 30, analizando todos los aspectos.

3.995 ptas.
Incluye CD-ROM.



Curso completo de mecanografía que ofrece la posibilidad de establecer una base firme para noveles, así como una perfecta corrección de "manías" de escritura para los más avanzados.

2.995 ptas.
Incluye CD-ROM.



CD-Rom que recoge alrededor de 700 recetas de cocina tradicionales. Cuenta con cálculo energético de calorías, valores nutritivos de cada alimento, etc.

2.495 ptas.
Incluye CD-ROM.



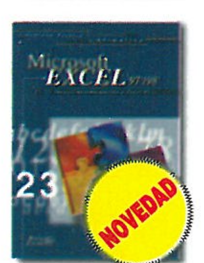
La solución ideal para agentes de seguros que incorpora la gestión completa de pólizas, la creación de informes a medida y una contabilidad integrada en la gestión de pólizas.

2.995 ptas.
Incluye CD-ROM.



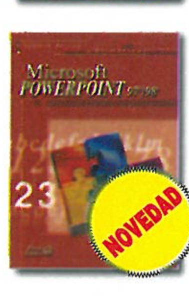
Word es un programa orientado a facilitar la escritura de documentos. A lo largo de este libro se inicia al lector en esta herramienta para, capítulo a capítulo, hacer de él un experto en su dominio.

2.995 ptas.
Incluye CD-ROM.



Excel 97 es la nueva hoja de calculo de Microsoft. Esta obra pretende introducir al lector en excel para, paulatinamente, enseñarle todos los trucos y atajos que conocen los expertos.

2.995 ptas.
Incluye CD-ROM.



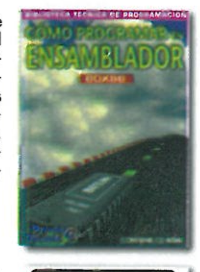
Esta obra ayuda al lector a conseguir que sus ideas resulten más atractivas y sirvan de plataforma para mostrar sus objetivos de una forma más llamativa.

2.995 ptas.
Incluye CD-ROM.



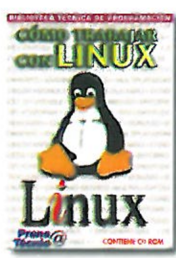
Introduce al lector de forma sencilla en el mundo de la programación de videojuegos, utilizando las técnicas más avanzadas en este campo. Recorrido por la historia de los videojuegos.

2.995 ptas.
Incluye CD-ROM.



Enseña paso a paso el mundo de la programación en Ensamblador. El CD-Rom está repleto de utilidades, programas, herramientas y tutoriales relacionados con el lenguaje Ensamblador.

2.995 ptas.
Incluye CD-ROM.



Explica cómo comenzar a usar Linux y será de utilidad para sacar más partido a su instalación. El sistema ofrece Multitarea, potente entorno gráfico, alto rendimiento y conectividad.

2.995 ptas.
Incluye CD-ROM.



DELPHI se puede considerar como uno de los entornos de programación visuales más poderosos y, a la vez, más fáciles de utilizar y aprender. Esta obra está orientada a nivel de principiante e intermedio.

2.995 ptas.
Incluye CD-ROM.



C es el lenguaje de programación por excelencia, el más utilizado para todo tipo de aplicaciones. Este libro le introducirá en la programación C de una manera clara y sencilla.

2.995 ptas.
Incluye CD-ROM.



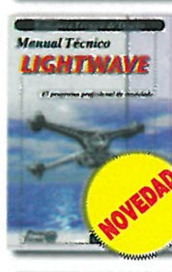
Dada su sencillez se trata del libro ideal para principiantes, así como en el lenguaje idóneo para las aplicaciones de Internet. El CD-ROM está repleto de aplicaciones para que los lectores consigan programar en Java.

2.995 ptas.
Incluye CD-ROM.



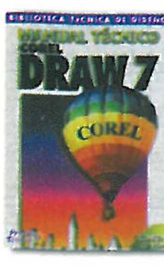
Sin duda alguna, uno de los campos más interesantes de la programación es el apartado gráfico. Hoy día las posibilidades de un PC en este campo son asombrosas, y con sólo unas líneas de código C o Ensamblador.

2.995 ptas.
Incluye CD-ROM.



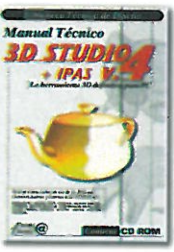
Introduce al lector en el mundo del modelado con Lightwave de una forma sencilla y práctica: Creación de efectos atmosféricos, animaciones... Incluye ejemplos prácticos.

2.995 ptas.
Incluye CD-ROM.



Ayuda al lector a dar sus primeros pasos con CorelDraw, con ejercicios prácticos para ampliar conocimientos en cada capítulo, con el fin de sacar todo el jugo al programa de diseño vectorial más utilizado.

2.995 ptas.
Incluye CD-ROM.



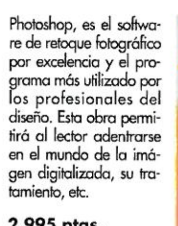
3D Studio ha sido durante años el programa de diseño más utilizado. Con esta obra el lector aprenderá a dominar la herramienta. Incluye un CD-ROM modelos, utilidades y texturas para 3D Studio.

2.995 ptas.
Incluye CD-ROM.



Photoshop, es el software de retoque fotográfico por excelencia y el programa más utilizado por los profesionales del diseño. Esta obra permitirá al lector adentrarse en el mundo de la imagen digitalizada, su tratamiento, etc.

2.995 ptas.
Incluye CD-ROM.





Descubre la última versión de Autocad 14, un programa de dibujo de propósito general, mayoritariamente difundido en el Ámbito del Diseño Asistido por Ordenador.

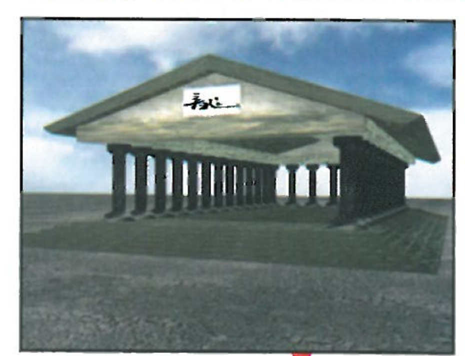
2.995 ptas.
Incluye CD-ROM.

Disfrute de descuentos al adquirir 2 o más productos • Consulta gratuita de atención al cliente • Los mejores productos del mercado al mejor precio •



PRODUCCIÓN NACIONAL

Otra pequeña muestra de los trabajos que cada mes nos enviáis a la redacción. Se nota que el nivel español en 3D no tiene nada que envidiar a los foráneos, y desde aquí os animamos a que sigáis así por mucho tiempo.

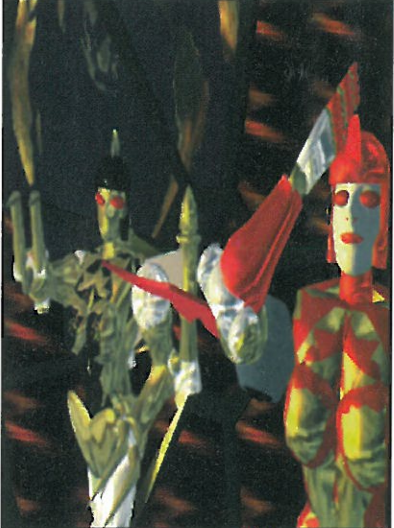


Título: TEMPLO
Autor: David López, de Madrid

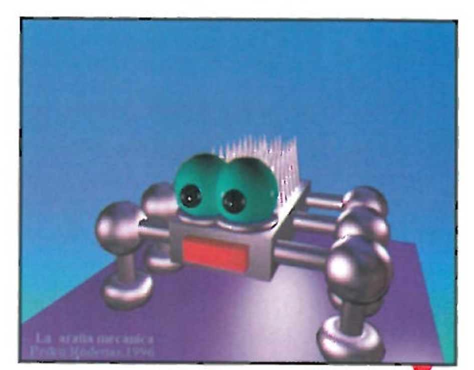




Título: GHOST IN THE SHELL
Autor: Ricardo Cabello, de Barcelona
Software: 3D Studio MAX 2.0



Título: SIN TÍTULO
Autor: Javier Medina, de Las Palmas de Gran Canaria
Equipo: 486DX2 a 66MHZ con 8MB de RAM
Software: 3D Studio 3.0(versión limitada), Paint Shop Pro Shareware



Título: LA ARAÑA MECÁNICA
Autor: Pedro Ródenas, de Barcelona
Hardware: Pentium 100 MHz con 40 Mb RAM
Software: 3D Studio 3.0



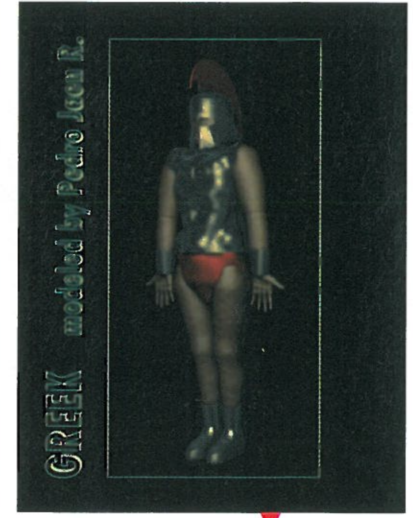
Título: NÉMESIS
Autor: Enrique Orrego, de Cádiz
Harware: Pentium 166 MMX con 72 MB de RAM
Software: Realizado con 3D Studio 3.0



Título: HAGAN JUEGO
Autor: Joaquín Quer, de Castellar del Vallés (Barcelona)
Hardware: Pentium 200 MMX, 128 MB de RAM
Software: Lightwave 5.5



Título: THE ATTACK OF THE MUTANT SPIDERS FROM MARS
Autor: Angel Lanza, de Cantabria



Título: GRIEGO
Autor: Pedro Jaén, de Córdoba
Hardware: Pentium 75 con 24 MB de RAM.
Software: 3D MAX 1.2, 3DS4, Metareyes 2, Rhinoceros y Paint Shop Pro

Contenido CD ROM

Este mes, en 3D WORLD tenemos el mejor CD-ROM que se puede encontrar, un compacto repleto de demos, Shareware, objetos, Plug-ins, efectos y demás software de utilidad para los amantes del 3D y la Infografía.

Nuestro recorrido comienza por los mundos virtuales que podremos crear con Animatek World Builder, uno de los mejores generadores de terrenos y paisajes disponibles en el mercado, con el cual daremos una nueva dimensión a nuestras escenas. Asimismo, incluimos una versión de prueba totalmente operativa de JetaReyes, el editor de gestos para 3D MAX que ya ofrecimos en su día para la versión 1.2 de MAX, y que, esta vez, regalamos preparada para funcionar en la versión 2 del modelador de Kinetix. También hemos incluido una versión Trial por un periodo de 30 días de Serif PagePlus, un sencillo programa de autoedición desarrollado por SPC, y que cuenta con avanzadas funcionalidades de autoedición, y una nueva Demo de Impulse, los creadores de Imagine, que ahora nos presentan su nueva estrella: Organica, un sencillo modelador que hará las delicias de nuestros lectores. Dentro del apartado para Mac ofrecemos las demos de Vertigo 3D Words, Specular 3D Web Workshop, PhotoFix y CyberStudio.

Y en primicia, y como apoyo al reportaje que le dedicamos en nuestras páginas, ofrecemos el compresor de vídeo Smacker, llamado a ser la alternativa a estándares actuales como AVI, QuickTime o MPEG, un revolucionario sistema de compresión de vídeo que dará mucho que hablar en los próximos meses.



Y como siempre, nuestras recopilaciones de Plug-Ins, objetos, IPAS, utilidades, ejemplos de los artículos de 3D WORLD y creaciones de los lectores, con un apartado especial dedicado al programa de modelado y animación Imagine, para el cual ofrecemos un directorio cargado de objetos, texturas, efectos y más utilidades para trabajar con este estupendo software. En definitiva, un CD-ROM en la línea de 3D WORLD, siempre ofreciendo lo mejor que se puede encontrar.

SOFTWARE INCLUIDO EN EL CD

DEMOS

PC
- Animatek World Builder
- Serif PagePlus
- JetaReyes
- Organica

MACINTOSH
- Vertigo 3D Words
- 3D Web Workshop
- PhotoFix
- CyberStudio

Edición de Vídeo
- Smacker

Imagine
- Brushmaps
- Efectos
- Objetos
- Texturas

IPAS para 3D Studio
- Catail
- Flame_i
- Count
- Pastel
- Puppet
- Ripple
- Xwave

Plug-Ins para 3D Studio MAX
- 3Dpoly
- Ccube
- Gray
- Helicoid
- Killer
- Objimp
- RGB

Plug-Ins Para Lightwave
- Acalpha
- Alphclip

- CurveToMotion
- Dbp2
- Encrypt
- LoadFromScene
- Ntavi
- Spiral
- Squash
- Taut
- Vdem2Stm

UTILIDADES

PC
- DirectX 5
- 3DStoPOV
- Xing MPEG Player
- PovCAD 4
- Video for Windows
- QuickTime

DUAL

Objetos
- 3DS (37)
- VRML (72)

Ejemplos de los artículos
- Workshop Programación
- Lightwave

JETAREYES



Ésta es una de las estrellas de este CD, que ya causó furor cuando en su día la ofrecimos para la versión 1.2 de MAX. Se trata del editor de gestos de REM Infográfica, un software que ya ha sido galardonado allí donde ha sido conocido.

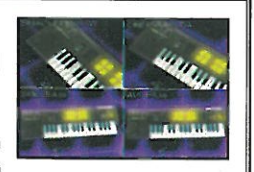
Para instalar la demo de JetaReyes tendremos que abrir el directorio JETAMAX2 del CD-ROM y, una vez allí, pinchar en el icono SETUP. El proceso de instalación es extremadamente sencillo, durante el cual tan solo tendremos que indicarle al instalador el directorio donde se encuentra ubicado 3D MAX 2. Una vez hecho esto, se procederá a copiar el Plug-in en nuestro disco duro y, para utilizarlo, tendremos que cargar el fichero FELIX(JETA).MAX, que se encontrará en el subdirectorio SCENES de 3D MAX 2, y abrir el panel de modificadores.

Junto con JetaReyes se incluye un manual en formato HTML dentro del directorio HTML, que podremos visualizar a través de Netscape o del Internet Explorer. Además, hay que asegurarse de que tenemos en el directorio SYSTEM de Windows las librerías OpenGL (que podremos encontrar en el directorio \UTILS\OPENGL del CD si no las tenemos instaladas) ya que, de lo contrario, no podremos ejecutar el editor de gestos JEDIT.

OBJETOS

Dentro del directorio OBJETOS (carpeta Objetos 3D en Macintosh) descubrimos 109 nuevos modelos, de los cuales 72 vienen en formato VRML y 37 para 3D Studio. Los hay de todo tipo, desde objetos comunes hasta modelos de animales, pasando por vehículos, objetos deportivos, naves espaciales y demás.

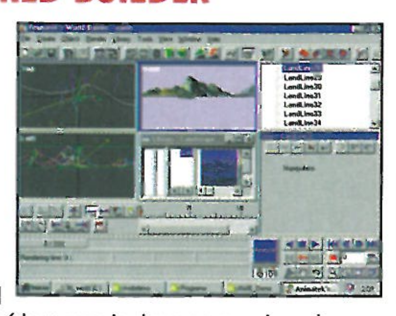
SMACKER



En el directorio SMACKER del CD-ROM se encuentra la versión completa de este estupendo y novedoso sistema de compresión de vídeo, del que hablamos en la sección de software de este mes. La instalación del compresor Smacker se realiza a través de un subdirectorio llamado SMACKER31, abriendo el icono SETUP, el cual dará paso a la instalación del programa.

Asimismo, en el mismo directorio se encuentra un subdirectorio llamado HTML que nos ofrece toda la documentación necesaria acerca de Smacker, así como los ejemplos realizados con Smacker que se han tratado en nuestras páginas.

ANIMATEK WORLD BUILDER



El plato fuerte de este CD corresponde al entorno de creación de paisajes y terrenos World Builder, de Animatek. Se trata de una demo limitada al tamaño del render, y con la cual podremos conocer las maravillas de este software, al igual que con las imágenes y vídeos que incluye como ejemplo.

Esta demo se encuentra en el directorio \AWBDEMO del CD-ROM, y para instalarla tan solo tendremos que abrir el icono "AUTORUN", con lo que aparecerá un menú del cual seleccionaremos la opción "Install", tras lo cual accederemos a un proceso de instalación de sobra conocido (seleccionar directorio de destino, grupo de programas a crear, etc...). Una vez instalado, podremos probar la Demo seleccionando el icono AWB20 dentro del directorio donde haya sido instalada.

Para saber más sobre World Builder u otros productos de Animatek, podéis poneros en contacto con Develon Data Systems, distribuidor del paquete, a través del teléfono (91) 534-82-80, o por Fax (91) 534-15-82.

SERIF PAGEPLUS



Dentro del directorio raíz del CD-ROM, si abrimos el icono SETUP, accederemos a la instalación de la versión Trial de Serif PagePlus, un programa de autoedición desarrollado por Software Publishing Corp, que ha sido cedida a 3D WORLD por SPC Ibérica.

Se trata de una versión de prueba operativa durante 30 días a contar desde la fecha de instalación, y una vez abierto el icono mencionado accederemos al instalador, el cual analizará la configuración del equipo y nos pedirá que introduzcamos el nombre de usuario y la empresa, tras lo cual comenzará la instalación. Lo primero que debemos introducir es si deseamos realizar una Ejecución desde CD Optimizada (la opción recomendada por el instalador), Ejecución desde CD Minimizada o Instalación en el disco duro (la opción que hará que el programa se ejecute más rápidamente). Una vez especificado, el programa nos pedirá que especifiquemos el directorio del disco duro donde se instalará la aplicación y el tipo de instalación (las ya conocidas instalaciones Típica, Completa o Personalizada), tras lo cual la Demo quedará instalada en el ordenador.

Si el lector desea más información sobre los productos de SPC, puede solicitarla a través del teléfono (91) 710-35-82 o Fax (91) 710-33-27

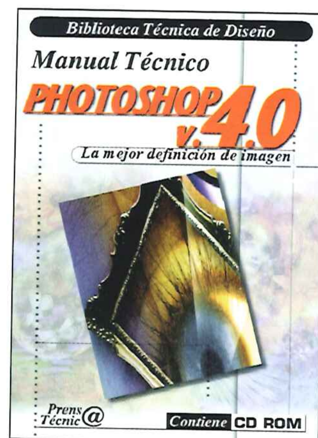
10 RAZONES para SUSCRIBIRSE a 3D World

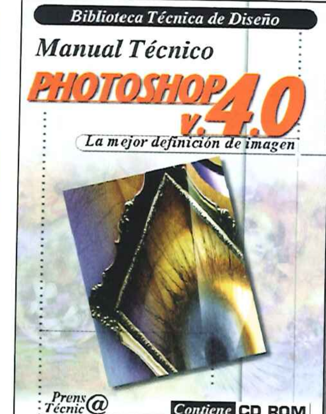
Si quieres saber todo lo que nunca te atreviste a preguntar sobre el mundo de las 3D... 3D WORLD es tu revista.

1. **Imprescindible** si quieres entrar en el mundo 3D, aprender de manera sencilla y sin esfuerzo el uso de las herramientas más utilizadas por los profesionales como 3D Studio, 3D Max, Lightwave, Caligari Truespace, Power Animator, etc.
2. **Si** ya tienes ciertos conocimientos podrás actualizarlos, mejorarlos y convertirte en un experto con los cursos básicos y secciones de trucos.
3. **Definitivamente** si eres un experto, 3D World es tu revista. Noticias, entrevistas, novedades del mercado, versiones de evaluación.
4. **Todos** los meses, de regalo, un completo CD-ROM, colección del mejor shareware 3D, modelos, herramientas, demos de programas comerciales, etc.
5. **Grandes** sorpresas durante todo el año 97.
6. **La** recibirás cómodamente sin moverte de casa.
7. **Descuentos** especiales a los suscriptores en promociones posteriores.
8. **Te** aseguras pagar el mismo precio durante todo el año.
9. **En** agosto, vete de vacaciones tranquilo. 3D WORLD llegará a tu buzón como siempre.
10. **Y** durante este mes, para todos los suscriptores dos libros con CD-ROM de regalo.

Está bien, esta vez va en serio, todos aquellos que acertéis suscribiéndoos a 3D World podréis elegir gratis dos super regalos de entre estos tres:

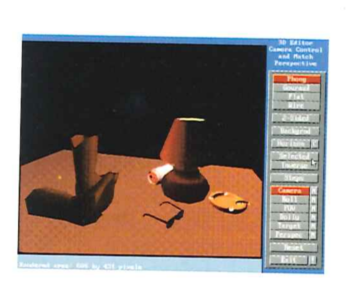






Manual Técnico de Photoshop v.4.0 (Colección Biblioteca Técnica de Diseño)

- Software de retoque fotográfico por excelencia.
- Programa más utilizado por los profesionales del diseño.



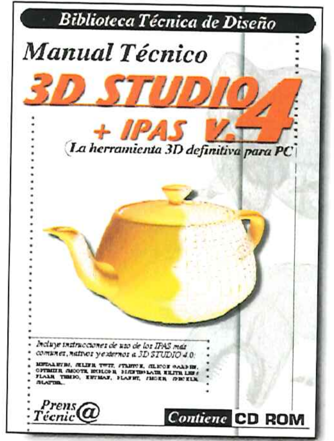


Manual Técnico de Autocad 14 (Colección Biblioteca Técnica de Diseño)

- Programa de diseño asistido por ordenador con diversas aplicaciones.
- Incluye colección de imágenes cedidas por la empresa Autodesk y dibujos de libre disposición de AutoCAD.

Manual Técnico de 3D Studio 4 e IPAS (Colección Biblioteca Técnica de Diseño)

- Todos los secretos de 3D Studio paso a paso.
- Explicaciones del uso de los IPAS más conocidos.
- Incluye CD-ROM con demo de 3DS, IPAS, modelos y texturas.



CONTENIDO DEL CD ROM

Este mes, nuestro CD-ROM nos lleva por los mundos virtuales que podremos crear con Animatek World Builder, uno de los mejores generadores de terrenos y paisajes disponibles en el mercado, con el cual daremos una nueva dimensión a nuestras escenas. Asimismo, incluimos una versión de prueba totalmente operativa de JetaReyes, el editor de gestos para 3D MAX que ya ofrecimos en su día para la versión 1.2 de MAX, y que esta vez regalamos preparada para funcionar en la versión 2 del modelador de Kinetix. También hemos incluido una versión Trial por un periodo de 30 días de Serif PagePlus, un sencillo programa de autoedición desarrollado por SPC, y que cuenta con avanzadas funcionalidades de autoedición, y una nueva Demo de Impulse, los creadores de Imagine, que ahora nos presentan su nueva estrella: Organica, un sencillo modelador que hará las delicias de nuestros lectores.
Y como siempre, nuestras recopilaciones de Plug-Ins, objetos, IPAS, utilidades, ejemplos de los artículos de 3D WORLD y creaciones de los lectores, con un apartado especial dedicado al programa de modelado y animación Imagine, para el cual ofrecemos un directorio cargado de objetos, texturas, efectos y más utilidades para trabajar con este estupendo software. En definitiva, un CD-ROM en la línea de 3D WORLD, siempre ofreciendo lo mejor que se puede encontrar.

3D WORLD
NÚMERO 15 DEPÓSITO LEGAL: M-2075-1997
Prens@ Técnic@

DEMOS
Pc Animatek World Builder
Serif PagePlus
JetaReyes
Organica
Mac Vertigo 3D Words
3D Web Workshop
PhotoFix
CyberStudio

Edición Vídeo
Smacker

Imagine
Brushmaps • Efectos • Objetos • Texturas

Objetos
3DS (37) • VRML (72)

IPAS:
3D Studio
Catail • Flame_i • Count • Pastel • Keydxf • Axpbox • Puppet • Ripple • Xwave

Plug-Ins:
3D Studio MAX
3Dpoly • Ccube • Gray • Helicoid • Killer • Maxdir • Objimp • R14DWG • RGB

Plug-Ins:
Lightwave
Acalpha • Alphclip • Bbox2 • CurveToMotion • Dbp2 • Doftoy • Encrypt • LoadFromScene • Ntavi • Rsvp_Trial • Spiral • Squash • Taut • Vdem2Stm

Utilidades PC
DirectX 5 • 3DStoPOV • Wcvt2pov • Polytrans • Paint Shop Pro • Xing MPEG Player • AcdSEE • Thumbs Plus • PovCAD 4

Ejemplos de los artículos
Workshop
Programación
Smacker
Lightwave
Recreación de Atmósferas

PC MAC

ANIMATEK WORLD BUILDER
Demo de este estupendo creador de terrenos y paisajes con el que conseguiremos mundos inimaginables para nuestras escenas.

JETAREYES 2
Versión totalmente operativa de JetaReyes, el sistema de edición de gestos de REM Infográfica, esta vez en versión para 3D MAX 2.

SERIF PAGEPLUS
Versión de evaluación operativa durante 30 días a partir de la fecha de instalación de este sencillo programa de autoedición de SPC.

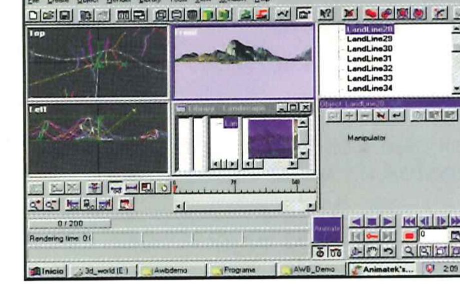
SMACKER
El nuevo sistema de compresión de vídeo que en poco tiempo plantará cara a los estándares actuales como AVI, QuickTime o MPEG.

ORGANICA
De la mano de Impulse, creadores, entre otros, del conocido imagine, nos llega ahora este sencillo pero potente programa de modelado para Windows.

OBJETOS
109 nuevos modelos en 3D, de los cuales 72 vienen en formato VRML y 37 para 3D Studio. Objetos comunes, vehículos, naves, aviones... nada escapa a nuestra recopilación.

PLUG-INS E IPAS
24 IPAS para 3D Studio y y 20 Plug-Ins para Lightwave, 2 para Real 3D y 9 para 3D MAX, con los que conseguiremos nuevos efectos al modelar.

ANIMATEK WORLD BUILDER. Uno de los mejores creadores de paisajes virtuales.



SERIF PAGEPLUS. Demo de este sencillo programa de autoedición para Windows.

JETAREYES. El mejor sistema de edición de gestos para 3D MAX 2.0.

